Software-Defined Lighting

by

Ye-Sheng Kuo

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering)
in the University of Michigan
2015

Doctoral Committee:

Assistant Professor Prabal Dutta, Chair
Professor David Blaauw
Associate Professor James W. Cutler
Assistant Professor Zhengya Zhang

2015

To my family and friends for their generous support.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Prabal Dutta, who is a brilliant, kind, and energetic scientist/engineer/educator. With his guidance, I have gained knowledge, technical skills, and learned perhaps the most important skill—*research*—along my Ph.D. journey. His endless ideas have inspired me to explore a wide space in my graduate study. He helped me to identify interesting and challenging problems, and taught me a systematic way to tackle the problems and evaluate tradeoffs between different approaches. In addition, I would like to thank my committee members, Professors David Blaauw, Zhengya Zhang, and James Cutler for their valuable feedback on my dissertation.

I could not have enjoyed working in Lab11 more, which is a great place with many phenomenal people. Ben Kempke, an RF ranging expert, who works alone and knows how to carve a pumpkin. Brad Campbell, a full-stack hacker and also an impressive carpenter. Pat Pannuto, a total system engineer who is involved in many different projects. Branden Ghena, a BLE device collector and ringleader of our undergraduates. Will Huang, an optimistic neighbor with a very different background. Noah Klugman, a mobile application developer with a truly artistic spirit. In addition, Sam DeBruin, Meghan Clark, Thomas Zachariah, Rohit Ramesh, Josh Adkins, Genevieve Flaspohler, and Neal Jackson form the greatest lab in the CSE department at Michigan—Lab11. I would to thank all my colleagues for sharing their knowledge, technical support, random chatting, and overall making Lab11 a better place to work. Among all colleagues, I would like to especially thank Pat Pannuto, who is also a brilliant scientist, and winner of many scholarships. I have worked with him closely on many projects, and he has demonstrated his excellence in both thinking and doing.

The winter in Ann Arbor would be much colder and lonelier without friends. Fortunately, I am lucky enough to have many good ones, especially tea time friends who made Ann Arbor a little warmer, enjoyable, and more like a home.

Lastly, I want to thank my significant other—Jie-Yu—and all my family members: Li-Chun, Yi-Sung, Chien-Pang, and Hsun-Chih for their endless support, love, and for always having faith in me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Software-Defined Lighting

by

Ye-Sheng Kuo

Chair: Prabal Dutta

For much of the past century, indoor lighting has been based on incandescent or gas-discharge technology. But, with LED lighting experiencing a 20x/decade increase in flux density, 10x/decade decrease in cost, and linear improvements in luminous efficiency, solid-state lighting is finally cost-competitive with the status quo. As a result, LED lighting is projected to reach over 70% market penetration by 2030. This dissertation claims that solid-state lighting's real potential has been barely explored, that now is the time to explore it, and that new lighting platforms and applications can drive lighting far beyond its roots as an illumination technology. Scaling laws make solid-state lighting competitive with conventional lighting, but two key features make solid-state lighting an enabler for many new applications: the high switching speeds possible using LEDs and the color palettes realizable with Red-Green-Blue-White (RGBW) multi-chip assemblies.

For this dissertation, *we have explored the post-illumination potential of LED lighting in applications as diverse as visible light communications, indoor positioning, smart dust time synchronization, and embedded device configuration, with an eventual eye toward supporting all of them using a shared lighting infrastructure under a unified system architecture that provides software-control over lighting.* To explore the space of *software-defined light-*

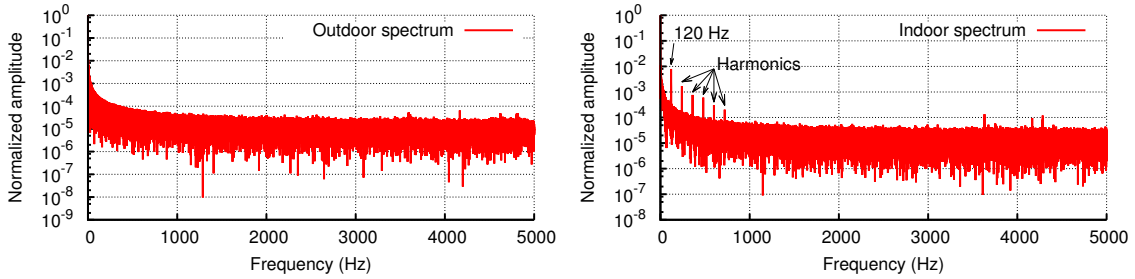
ing (SDL), we design a compact, flexible, and networked SDL platform to allow researchers to rapidly test new ideas. Using this platform, we demonstrate the viability of several applications, including multi-luminaire synchronized communication to a photodiode receiver, communication to mobile phone cameras, and indoor positioning using unmodified mobile phones. We show that all these applications and many other potential applications can be simultaneously supported by a single lighting infrastructure under software control.

CHAPTER 1

Introduction

Incandescent lights have played an important role in indoor lighting for the past one and half centuries. Today, incandescent lights continue to dominate the residential lighting market, primarily because of their warm color temperature and good color rendering. On the other hand, commercial lighting is mainly provided by fluorescent lamps, which offer longer lifetimes and higher efficiency than incandescent bulbs. Recent advances in solid-state lighting (SSL), however, have made LED-based solutions competitive for general lighting markets. Driven by Haitz's Law, which states that the cost per lumen of LEDs decreases by 10x per decade and the light generated per unit area of silicon increases 20x per decade [37], today's commercial LED lighting provides over 100 lm/W efficacy—higher than nearly any other technology—and a 50,000 h lifetime—longest among all lighting technologies [11, 63]. In addition to long lifetime and high efficiency, LED lights offer compact form factors, high color rendering index (CRI), high performance in low temperature, and require no hazardous mercury in their production. As the total cost of ownership (TCO) for LEDs achieves parity with conventional lights, they are gaining market acceptance, but the real potential of LED lighting has been barely explored.

One key difference between LED and other lighting technologies is that LEDs are capable of rapidly switching state—much faster than humans can perceive. Hence, high frequency-modulated LED lighting enables a visible light communication channel that is imperceptible to humans. The real opportunities for visible light communications (VLC) are enabled by the unregulated visible light spectrum and the widespread coverage of lighting infrastructure. Figure 1.1 shows the spectrum usage in an office illuminated by fluorescent lights. The largest peak occurs at 120 Hz, which is twice the AC mains frequency. Since the AC voltage waveform encounters two zero crossings per cycle, fluorescent lights oscillate at that rate. More importantly, the harmonic peaks are also narrow band, showing that the optical spectrum is largely unused and unpolluted. In addition, the Energy Independence and Security Act (EISA) of 2007 prohibits importing or manufacturing lights that fail



(a) Outdoor spectrum

(b) Indoor spectrum

Figure 1.1: Optical power spectrum of an outdoor environment and a typical fluorescent light-lit office space. The indoor spectrum is largely unused, with the exception of narrowband spikes at 120 Hz and their harmonics (due to AC zero crossings that cause the fluorescent light to flicker). Both indoors and outdoors, the visible light spectrum is largely unused, suggesting that it is a rich medium for new communication and other technologies.

to meet minimum efficiency and lifetime targets [84]. As a consequence of this legislation, the LED market share for general lighting is projected to reach 73.7% by 2030 [64].

1.1 Post-Illumination Opportunities for Lighting

As global LED lighting adoption increases, many post-illumination lighting applications in health, energy, entertainment, communications, indoor positioning, device configuration, and time synchronization, among many others, as shown in Figure 1.2 will emerge. Here, we identify some current opportunities, review research efforts already underway, and speculate on potential future applications.

Health. Light exposure to the eyes is the strongest effector of entrainment, or synchronization, of the internal circadian rhythm with the external environment. The timing, length, intensity, and wavelength of light to which one is exposed all affect entrainment [10, 23]. We envision a future in which smart badges capture individual exposure parameters [65], communicate them to the surrounding environment, and securely task nearby lights [1] to adjust spectrum and intensity to compensate for poor exposure, complementing approaches like the University of Michigan-developed Entrain mobile app [44].

Energy Efficiency. Lighting accounts for 26% of electricity consumption in commercial buildings and 19% of total electricity use in the U.S [63, 85]. Studies have shown that giving occupants fine-grained control over their own lighting can lead to substantial energy savings [18, 48]. Similarly, automated control systems could use mobile phones to detect ambient light levels and provide closed loop feedback to lighting control systems. These control systems could also adjust the lighting level dynamically in response to daylight,

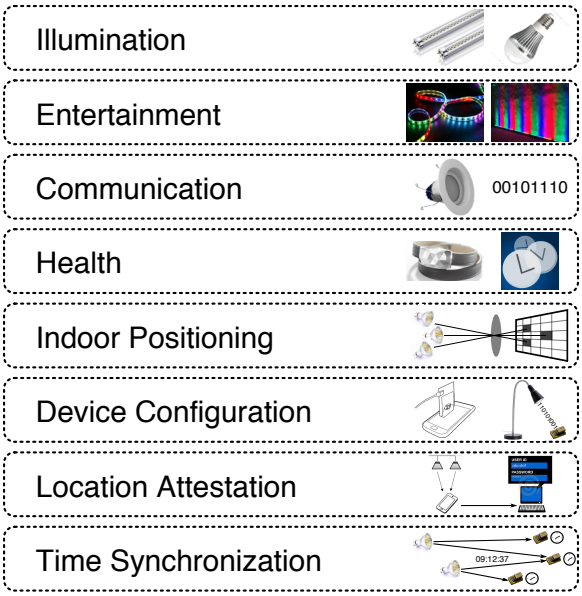


Figure 1.2: Potential applications of software-defined lighting.

saving even more energy. In addition, user location/profiles can be integrated in control loops to provide human tracking, personalized lighting environments, and energy savings.

Entertainment. Light intensity and color can be controlled by end users, with their mobile phones or programmatically,[1] to allow people to tailor their environments for fun and whimsy. Taken further, one might imagine lighting that flashes with the ambient music's rhythm and changes color or intensity in response to its spectrogram; future videos could include a "lighting track" akin to current sound tracks to enhance viewing experience.

Communications. LED lighting provides an additional communication channel to mobile devices or other dedicated or multiuse receivers. LED lighting can be used in low-rate data broadcasting or simply to provide network connectivity in locations where RF is unavailable. Notifications or alerts can be integrated with security systems to provide visible alarms for toxic gas or insecure windows. Semantic localization—room- or region-level indoor localization—is enabled by broadcasting data from lighting luminaires and receiving them with cameras integrated into mobile phones [2, 73]. This allows one to establish accurate location context without custom hardware or complex environmental modeling. For example, imagine a student pulls out their phone to reserve the conference room in which they are standing, simply by pressing a button that reserves whichever room the phone detects it is currently located in, secured from misuse by location attestation.

Location Attestation. For some applications, like lighting control, proof of presence is important. This ensures, for example, that only people who are physically present can

[1] http://blinken.eecs.umich.edu

control the lights in a classroom. To enable such scenarios, we propose location attestation as a service available to cloud or mobile applications. A mobile application that wants access to a room's lights would have to first prove its presence in that room by responding to an optical challenge transmitted by the room's lights. This service would take challenge-response authentication into the physical realm.

Beyond Semantic Indoor Positioning. For many years, researchers have been solving the indoor positioning problem using various range-based or range-free proximity techniques. Most proposed solutions focus on RF fingerprinting, proximity, and ranging; however, RF-based techniques suffer from channel variation, limited bandwidth, or intensive calibration, and they deliver location accuracy measured in meters. Leveraging visible light, we are able to show and order of magnitude improvement, providing decimeter-accurate indoor location estimates (§6).

Device Configuration. As sensors shrink in size and grow in number, conventional techniques for programming and configuration of devices become impossible. One possibility for programming and configuration is to employ broadcast optical interfaces, rather than electrical ones. For example, the Electric Imp is a system in an SD card form factor that integrates a microcontroller and WiFi networking. The WiFi configuration (SSID and password) is sent to the Electric Imp optically using a dedicated mobile app [31]. For an even smaller example, the Michigan Micro Mote (M^3) is a modular system with a microcontroller, radio, camera, energy harvester, and sensors in a 1 mm^3 form factor [56]. This "smart dust" system is self-sustaining and nearly invisible. Due to the minute form factor, traditional programming methods that rely on physical connection are difficult or impossible. For this reason, M^3 includes an optical frontend for programming [43].[2]

Time Synchronization. Highly-stable, pico- and nano-power clocks are beyond the state-of-the-art, so energy constrained millimeter-scale computers currently keep poor time. Hence, communications guard bands dominate the energy budget of low-power, wireless sensors. Researchers have demonstrated a sub-nW optical wake-up receiver whose standby power is three orders of magnitude lower than state-of-art RF wake up radios [43]. Such optical wake-up receivers can stay on for entire guard bands or even be always-on without depleting batteries. Luminaires could be tasked to program/wake-up sensors simultaneously for high-power RF communication, dramatically reducing guard time, providing energy-efficient global time synchronization and precision coordination of global tasks.[3]

[2] http://youtu.be/OM8WgnhcyOo
[3] http://eecs.umich.edu/~prabal/videos/ElectricImp+SDL.mov

1.2 Architecture for Software-Defined Lighting

To experiment with and evaluate some of the opportunities envisioned in §1.1, we propose a new architecture for indoor lighting systems—software-defined lighting (SDL). Our goal is to evolve these initial ideas toward a future architecture that can support these, and other, applications. Many of the arguments for rethinking the role of lighting rely on the fact that lighting is ubiquitous, so it provides full coverage indoors. Few have tried to argue for a stovepipe approach in which new lighting is deployed for each new application, since that would defeat the economies of scope that a single, shared infrastructure could provide. Given the strong desire to multiplex the future lighting infrastructure for a multitude of applications, one major goal of software-defined lighting must be to allow sharing both at the physical layer, and logically at the link and higher layers of the network stack.

Before we can reason about how to share the lighting infrastructure, we must understand how the new infrastructure will be used in potentially myriad different ways, and beyond today's state-of-the-art. Indeed, "for the past few years, visible light communications has been an active research topic, and research has already brought the technology closer to commercialization and standardization."[4] But much of this effort has focused on the visible light communication channel itself—or transmitter/receiver pairs—in which it was assumed that both ends of the link could be controlled and custom-designed to satisfy compliance with a standard. Unfortunately, this may not provide sufficient flexibility to incorporate the wider range of applications we envision for lights, in addition to the primary illumination role.

Rather than take a prematurely-standardized approach to the narrower problem of visible light communication, we advocate a more flexible approach to the broader opportunity offered by *software-defined lighting*. Our goal is to first *let chaos reign* and only then *rein in the chaos.*

Today, the mobile communications community's research goal is to "move beyond VLC links and to explore VLC-enabled networks, systems, and applications."[4] To enable these explorations through research-enabling research, we propose the architecture shown in Figure 1.3. This architecture provides the needed experimental flexibility for the research community but it is grounded in decades of investments in Internet technologies and more recently emerging concepts from the software-defined networking community.

[4] http://www.networks.rice.edu/VLCS-2014/

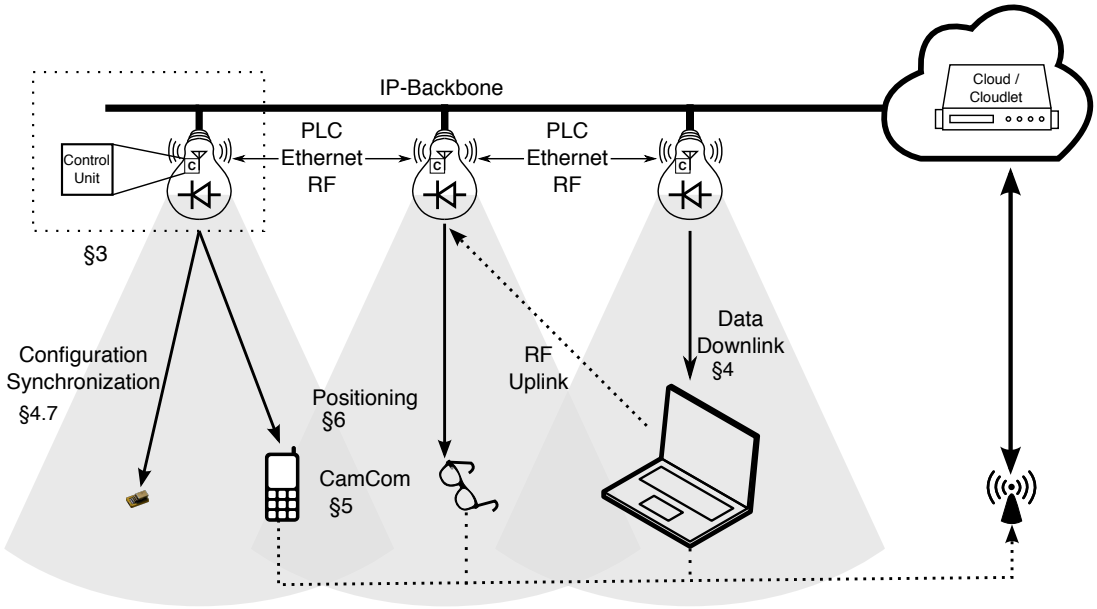


Figure 1.3: Proposed architecture for software-defined lighting. The architectural elements include: (i) luminaires (the lights themselves), (ii) a Internet Protocol (IP)-based backbone network running over radio frequency (RF), powerline carrier (PLC), or Ethernet that connects the lights to enterprise IP networks, (iii) cloud or cloudlet servers that manage and control the lights and offload computation for local clients, and (iv) receivers like sensors, smartphones, computational eyeglasses, and camera-equipped laptops that request and/or receive data and computational resources.

1.3 Design Challenges Across the System Stack

Although SDL presents many opportunities for new applications, it also raises many new challenges across the system stack. While some of these challenges may be apparent from the outset, many others will not reveal themselves until experimental systems are designed and deployed. We anticipate a number of challenges, as described below, and expect others will emerge as the proposed research unfolds.

Cost and Value. For eventual adoption, either the cost of smart lighting must become marginal or its value must become substantial—in the form of new applications enabled because of the lighting system. Practically, this means minor modifications of commercial products will be more palatable to adoption, but significant functional enhancements that enable new revenues streams could also emerge. The challenge lies in determining how to best leverage infrastructure to support applications that provide value to the owners or occupants of indoor spaces without inhibiting the exploration of radical ideas. These must be identified, prototyped, and evaluated to make smart, software-defined lighting a commercial reality, and the value of applications must exceed the cost of deploying them.

Flexibility. Balancing long-term cost and value concerns with the flexibility needed for immediate experimentation poses a tradeoff. Specifically, we need a powerful enough platform to test ideas but a clear pathway to inexpensively realizing them. To explore this, we argue for a multi-pronged approach: pushing the extremes in power and performance of solid-state lighting transmitters and receivers, from IEEE 802.15.7 standards-compliant technology to nano-watt smart dust receivers with custom circuits and realizations based on slight modifications to existing products to ground exploration.

Networking. Many of the envisioned applications for software-defined lighting require dynamic information to be received (and subsequently transmitted) by the lighting infrastructure. How should this data be transmitted to the lights in the first place? Many options exist, including Ethernet, PLC, and RF, but they occupy different points in the space of cost, complexity, and data rate. Both Ethernet and PLC provide networking and power delivery, reducing wire count. However, many existing buildings do not have ubiquitous Ethernet ports, requiring costly infrastructure upgrades to support networking. On the other hand, PLC provides lower data rates than Ethernet, and network connectivity depends on the layout of the power distribution network. Commercial buildings often have many electrically distinct domains, separated by transformers that isolate PLC signals. Isolated PLC nodes may require PLC repeaters, RF links, or Ethernet/PoE support, to form large, complete networks. This suggests that a multihop, multi-link protocol like IP may be ideal. Longer term, if high data rate applications are deployed, high capacity backhauls must also be built.

Synchronization. In some cases (e.g. with cameras as line-of-sight receivers), it is possible for multiple lights to transmit different data through optical modulation since each pixel in the image plane corresponds to a different region in the world view. In other cases, diffusing receivers (e.g. photodiodes) will mix incoming light from multiple sources, so either the transmissions must be time synchronized or they must employ some form of orthogonality, like TDMA or FDMA. This creates a space of tradeoffs in protocol complexity, computing requirements, and communications efficiency.

Modulation. For applications that require lights to transmit information optically, how should the data be encoded and modulated? The answer depends in part on the receiver, in part on the illumination, and in part on any applicable standards. Although VLC standards like IEEE 802.15.7 exist, they make many assumptions about the available infrastructure and assume control over receivers. This makes adoption challenging as it requires two different industries—lighting and consumer electronics—to agree to absorb costs without any clearly-defined market opportunity. A different approach, which we advocate, makes physical layer modulation schemes programmable. Today, we can leverage existing and emerging receivers, like smartphones and smart dust, respectively, and tailor visible light

communications to each, exploring opportunities before market forces conspire to make them profitable. Tomorrow, we can decide which modulations schemes are best given the computational, communications, and energy resources available for transmitters and receivers.

Multiplexing. A central requirement of reusing the lighting infrastructure for other applications is that these applications must share the lights, through some kind of multiplexing, to support the various applications without compromising on the lighting's primary role as illumination. The questions are not limited to the physical layer, however. For example, in applications that require spatial control over lights by users, how should different principals (applications, services, or users) control the lights and resolve conflicts? This raises issues of access control.

Backward-Compatibility. Software-defined lighting must offer a modicum of compatibility with existing lighting systems, while new functions should support incremental deployment rather than require wholesale infrastructure replacement. Practically, this means that preference should be given to platforms that can replace Edison, can, or fluorescent lamps for immediate adoption, ensuring that these platforms can be used both for immediate lighting and alternate uses.

Although software-defined lighting presents many interesting opportunities for new applications, it also raises many new challenges across the system stack. While some of these challenges may be apparent from the outset, many others will not reveal themselves until experimental systems are designed and deployed. We identified a number of challenges, as described below, and expect others will emerge as the proposed research unfolds.

1.4 Thesis Statement

The high switching speed and wide color palette of solid-state lighting enables a diverse array of post-illumination applications that can be simultaneously realized by a shared lighting infrastructure, under software-defined communications and control, without adversely affecting the primary illumination goal of indoor lighting.

1.5 Contributions of this Dissertation

We have designed, implemented, and evaluated an architecture for, and applications of, software-defined lighting, thus providing guidance to this emerging research area with its

first workshop[5] scheduled in 2014, being held with "the objective of...moving beyond VLC links...to explore VLC-enabled networks, systems, and applications." Our vision of SDL has been presented in VLCS'14 with Pat Pannuto, and professor Prabal Dutta [49].

In §3, we present a modularized architecture of SDL. That architecture is composed of an LED Frontend, Control Plane, and Data Plane. We explore the design spaces for each subsystem, identify the building blocks for low-rate applications, and implement a fully custom SDL setup that receives data from Power Line Communication (PLC) which is passed to a local Control Plane for decision making and is ultimately optically transmitted through an LED Frontend. In addition to this modularized system, we implement an integrated development platform that marries control and data plane. This platform was presented in HotNets '10 and SenSys '12, and is joint work with professor Prabal Dutta, professor Akos Ledeczi, professor Thomas Schmid, Dr. Peter Volgyesi, and Pat Pannuto [28, 51]. We also demonstrate the design of LED transmitters and receivers that support a variety of applications and show how commercially-available LED luminaires can be easily modified, often with the addition of just a transistor and microcontroller/oscillator, to support many of these applications. We also present several receivers, both custom and commercially-available, including two visible light communications (VLC) receivers, a third-party "smart dust" receiver that can be programmed optically, and a smartphone receiver that uses its CMOS camera to receive data from lights.

In §4, we build upon the flexible SDL platform to explore microsecond-level synchronized transmissions from multiple LED lights over a multihop *wireless* backbone. This synchronization level is necessary to avoid intersymbol interference in optical downlinks, and it is achieved using a new technique called *Floodcasting*. Floodcasting is a data dissemination, time synchronization wireless protocol that is designed using concurrent transmission. We mathematically derive the macro behavior of receiving concurrent transmitted packets, and show that carrier frequency mismatch lowers packet reception rate (PRR). We present two methods—continues automatic gain control and automatic frequency compensation to mitigate the carrier frequency offset. These techniques are presented in SenSys '12, and this is a collaborative work with Pat Pannuto, professor Thomas Schmid, and professor Prabal Dutta. In Floodcasting, nodes alternate receiving and transmitting data, but unlike traditional relay networks, transmissions occur with sub-symbol timing jitter, allowing multiple nodes to concurrently transmit data over the wireless backbone, and crucially, to optically transmit the data with tight synchronization. This allows simple receivers to receive transmissions from multiple LED sources without destructive interference or receiver-based decoding techniques.

[5] 1st ACM Workshop on Visible Light Communication Systems–September 7, 2014

In §5, we consider the problem of communicating data between the lighting infrastructure and unmodified smartphone cameras. We demonstrate that it is possible to transmit data at much higher than a camera's frame rate when the camera uses a rolling shutter. We show that simple frequency tones in the 1 to 10 kHz range are detectable and discernible by smartphone cameras while being imperceptible to humans. By rapidly alternating between different tones, it is possible to implement frequency-shift keying transmitters, and by Manchester encoding the data, it is possible to transmit data packets to smartphones. Although our techniques require substantial computational resources and operating system support that is not yet widely available, these capabilities are expected to soon appear in the marketplace. Moreover, significant efficiency gains may be possible with additional optimization of the algorithms.

In §6, we explore a very challenging problem—indoor positioning—and show how by using a combination of camera communications and photogrammetry it is possible to obtain indoor position *and* orientation with an order of magnitude better accuracy than the state-of-the-art RF-based solutions, and four times better than optical techniques that employ custom receiver hardware. This is the first work to demonstrate that indoor positioning using visible light is possible and almost practical. The visible light positioning system, named Luxapose, is presented at MobiCom '14, and it is a joint work with Pat Pannuto, Ko-Jen Hsiao and professor Prabal Dutta [50].

In §7, we tie all of the pieces of a software-defined lighting architecture together into a single testbed and deploy four different applications on the shared lighting infrastructure. This testbed demonstrates that it is possible to combine a range of different luminaires (both COTS and custom), a range of different receivers (smartphones, smart dust, commercial products, and custom frontends), and a range of different data planes (powerline communications, Ethernet, and 802.15.4 wireless) into a coherent system controlled by a single server that offers access to multiple clients.

In §8, we summarize our results and propose some possible directions for future work. The first modularized SDL prototype we implemented is limited to PLC data plane. We would expect to explore more options including an integrated system that supports Ethernet with PoE. We implemented Floodcasting in an FPGA, which is a more flexible and powerful platform than the one we imagine eventually appropriate for a low-cost, largely-unified SDL system. We also need to explore alternative approaches to synchronize luminaires that are more compatible with conventional techniques. For example, side channels (GPS), Precision Time Protocol via Ethernet (IEEE 1588), or PLC broadcast might offer viable solutions. Another area that deserves attention is improving the performance of some applications. For example, the current implementation of our visible light positioning system

is computationally expensive, and it could be potentially optimized by more efficient image processing algorithms or more on-phone pre-processing. Finally, deploying our SDL vision at larger scales would enable us to evaluate many other post-illumination applications on a shared lighting infrastructure, test our architectural and design choices, establish the viability of our approach, and determine the flexibility of our service abstractions and APIs.

CHAPTER 2

Background and Related Work

Lighting in residential and commercial buildings accounts for nearly 19% of total electricity consumption in the U.S. [89]. In the U.S., over 8.2 billion lamps are installed in residential, commercial, industrial, and outdoor environments. The majority of electricity consumption can be attributed to commercial buildings (49.8%) [63]. Figure 2.1 shows the market share of lighting technologies in each environment. Although the adoption rate of LED technologies has increased significantly in the past decade, LED market share is still below 1% in 2010. Today, 22% of electricity is consumed by inefficient incandescent lights. Recently, some countries have passed legislation aimed at phasing-out inefficient incandescent bulbs. Starting in 2014, manufacturing/importing 40 W and 60 W incandescent light bulbs has been prohibited in the U.S. In part in response to this policy, the market share (% of lm-hr) of LED is projected to increase to 9.5% by 2015 and 73.7% by 2030 [64].

The main advantages of solid state lighting (SSL) is higher efficacy (measured as lumen/watt (lm/W)) and longer lifetime. The efficiency of LED lighting is ˜8X better than incandescent light. The lifetime of typical LED lights is up to 50,000 h, far greater than the typical ˜1000 h lifetime of incandescent bulbs [11]. Dimmability, controllability, smaller package, mercury reduction, low temperature performance, and high color rendering index (CRI) are other benefits provided by LED lights. A key feature of LEDs—high bandwidth—is enabling a new era in wireless communications—visible light communication (VLC). VLC provides a wireless communications channel that could be used on top of indoor lighting. The visible light spectrum is unlicensed and unregulated, and its high bandwidth allows for high speed data communication. Line of sight (LoS) propagation characteristics allow high spatial reuse of the medium. Adjacent sources that do not overlay can transmit data without employing any multiple access techniques. Simple VLC transceivers can also be low-cost, since they directly modulate a baseband signal. Unlike RF transceivers, VLC transceivers require no sophisticated circuitry for up/down

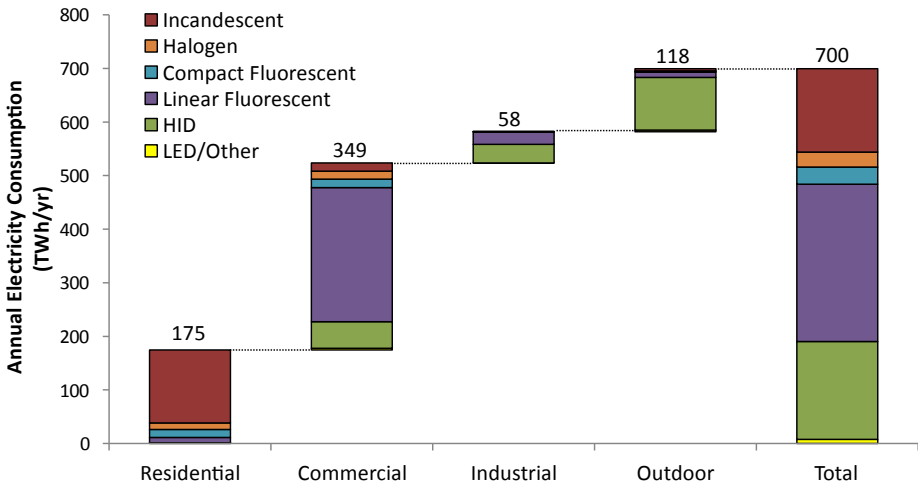


Figure 2.1: Annual electricity consumption in lighting fixtures in the U.S. [63].

conversion, reducing cost. VLC enables new approaches for low-cost, high accuracy indoor positioning as well. RF-based indoor localization, in contrast, suffers from multipath effects and limited bandwidth, making it difficult to accurately determine localization.

VLC on the other hand, has a number of drawbacks not shared by RF communication. First, VLC communication range is relatively short. Communication distance and available data rate are each functions of the signal to noise ratio (SNR). Both the configuration of the transmitter and any receivers (directed LoS or diffused light), as well as the distance, angles, and environment also significantly affect transmission distance and data rate. Second, VLC is prone to interference from ambient light. Natural sunlight or other artificial light could act as a source of interference. Although these interference are low frequency and could most probably be filtered out, strong direct sunlight can saturate a receiver. Therefore, indoor communication can be considered the more robust venue for VLC links. Thirdly, the VLC link is visible, so it cannot provide a communications link in a dark environment. Finally, transmitting signals using high-brightness luminaries draws significant power, leading to VLC links to low-power devices likely being unidirectional. VLC technologies are not an RF wireless replacement, but instead enhance wireless flexibility, improve aggregate bandwidth and can enable novel applications.

In this section, we discuss the LoS channel model, modulations for VLC, IEEE standard, and existing applications have been built on top of VLC.

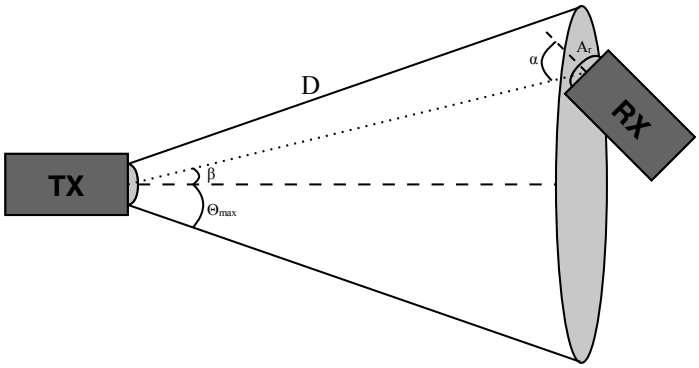


Figure 2.2: Line of sight (LoS) link model [22].

2.1 VLC Channel Model

Short range VLC link can be simply characterized using the line of sight (LoS) model. With LoS model, theoretical received power, path loss and link budget can be analyzed. Figure 2.2 shows the parameters in the model.

α Angle between transmitter-receiver direct line and receiver's normal line.

β Angle between transmitter-receiver direct line and transmitter's normal line.

D Distance between transmitter and receiver.

θ_{max} Transmitter's maximum radiate angle.

A_r Receiver's active area.

Path loss (L_L) is defined as received flux (F_r) divided by transmitted flux (F_t), which can be expressed in the following equation:

$$\begin{aligned} L_L &= \frac{F_r}{F_t} \\ &= \frac{I_0 \times g_t(\beta) \times \Omega_R}{I_0 \int g_t(\theta) \mathrm{d}\Omega}, \end{aligned}$$

where I_0 is the transmitter intensity, $g_t(\beta)$ is the transmitter's spatial distribution, Ω_R is a solid angle. The LED's spatial distribution often can be modeled using Lamert's cosine law [34] which is written as: $g_t(\theta) = cos^m(\theta)$. Where m is the order of Lambertian emission. m can be calculated as: $m = -ln2/ln(cos\phi_{\frac{1}{2}})$. Where $\phi_{\frac{1}{2}}$ is the angle that radiates half intensity relative to center ($\theta = 0$). For many commercial LEDs, $m = 1$ is a close approximation. Figure 2.3 shows the distribution ($g_t(\theta)$) of a commercial LED,

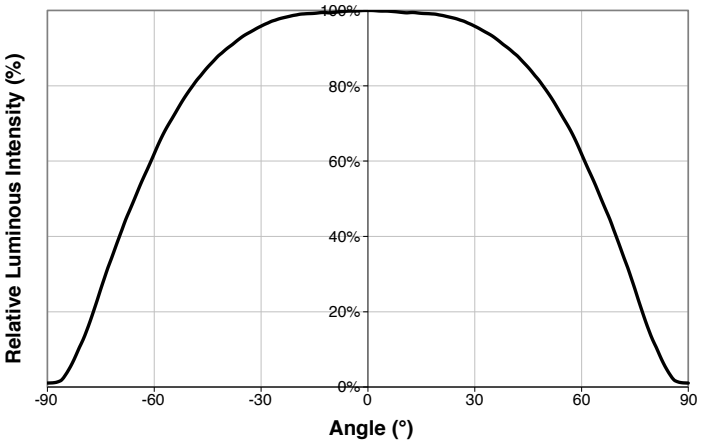


Figure 2.3: Spatial distribution ($g_t(\theta)$) of a commercial LED [21].

and $\phi_{\frac{1}{2}} \approx 66$. The solid angle ($\Omega_R$) can be approximated to $A_r cos(\alpha)/D^2$, and $\Omega = 2\pi(1 - cos(\theta))$ [22]. Hence, the path loss (L_L) can be simplified as:

$$\begin{aligned} L_L &\approx \frac{A_r cos(\alpha) g_t(\beta)}{D^2 \int_0^{\theta_{max}} g_t(\theta) \times 2\pi sin(\theta) \mathrm{d}\theta} \\ &\approx \frac{A_r cos(\alpha) cos^m(\beta)}{2\pi D^2 \int_0^{\theta_{max}} cos^m(\theta) sin(\theta) \mathrm{d}\theta} \\ &\approx \frac{A_r(m+1)}{2\pi D^2} cos^m(\beta) \times cos(\alpha) \end{aligned}$$

Received spectral density ($S_r(\lambda)$) can be derived from transmitted spectral density ($S_t(\lambda)$) multiplied by path loss (L_L).

$$S_r(\lambda) = S_t(\lambda) \times L_L$$

Received power (P_r) can be derived from integration of spectral density and spectral response of optical filter ($R_f(\lambda)$) and photodetector ($R_D(\lambda)$).

$$P_r = \int S_r(\lambda) \times R_f(\lambda) \times R_D(\lambda) \mathrm{d}\lambda$$

2.1.1 Ambient Light Interference

Ambient light affects performance of VLC by causing interference. The sunlight is a wide spectrum signal with very low frequency. It often can be decoupled by a high-pass filter. However, this filter is usually placed after transimpedance amplifier (TIA) in the processing

chain, so strong sunlight could still saturates receiver directly. In an indoor environment, dimmable incandescent and fluorescent light also produce noise. Shot noise and flickering from fluorescent light are two major sources or noise. Electronic ballasts can produce up to a 1 MHz flicker. Interference from ambient light deteriorates the signal to noise ratio (SNR), which lowers the communication distance and/or data rate.

2.2 Human-Friendly Modulation Schemes

VLC provides a wireless communication channel on top of lighting infrastructure. This channel violate the most basic primitive—lighting the room. Namely, modulation schemes shouldn't have any visible impact to human eyes. Human eyes have a integration time around 0.1 s [17], so light intensity should remain constant withing the window with high-speed flickering. The common practice is that the flicker period should be less than the maximum flickering time period (MFTP). The MFTP is the maximum period that humans cannot perceive a change in light intensity [74]. In general, 5 ms is considered safe. Hence, the frequency of modulation schemes should be at least 200 Hz [9]. Another desired property for modulation schemes is dimming support. A human pupil adjusts the amount of light perceived by eyes, the result of which has a square root relationship with measured light [75]. To support dimming, modulation schemes need to maintain data link under a wide range of lighting intensity.

IEEE 802.15.7 standard defines three modulation schemes. They are on-off-keying (OOK), variable pulse position modulation (VPPM), and color shift keying (CSK) [5]. Table 2.1 summarizes three modulation schemes. In addition to the modulation schemes defined in this standard, orthogonal frequency division multiplexing (OFDM) aka discrete multi tone (DMT) is a commonly used technique for high data rate VLC applications.

2.2.1 OOK

On-Off-Keying (OOK) uses two different lighting intensity to represent "on" and "off". However, pure OOK is not a flicker free modulation and doesn't support dimming. Source coding is necessary to balance the number of "0" and "1" symbols. Manchester or 8B10B codes can be used to achieve flicker free communication. Dynamic adjusting intensity of "on" and "off" supports dimming directly. However, adjusting intensity might cause color shift. Alternatively, dimming can be achieved by inserting compensation symbols (CS) in MAC layer. A compensation symbol is a special symbol in the packet that has different brightness level. By inserting CS in a VLC packet, the average brightness can be changed.

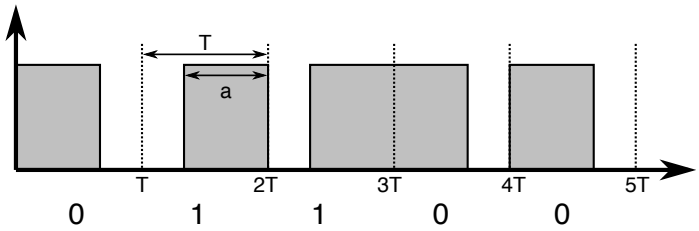


Figure 2.4: VPPM waveform. It uses position of pulse to represent symbol. It's equivalent to Manchester encoding signal. Transition from "1" to "0" stands for symbol "0" and vise versa. VPPM support dimming by varying duty cycle (a/T).

2.2.2 VPPM

Variable Pulse Position Modulation (VPPM) uses pulse positioning to represent symbols. VPPM is equivalent to OOK with Manchester encoding with an arbitrary duty cycle. Figure 2.4 shows the modulation waveform. It dims a light by varying the duty cycle (a/T). Compensation symbols could also be used in dimming support.

2.2.3 CSK

One method for generating white light is to mix red, green and blue. RGB could carry different information to increase data rate. Color-shift-keying (CSK) mixes 3 light sources out of 7 pre-defined colors to create symbols. The 3 light sources are vertex of a triangle on CIE 1931 color coordinates [40]. Through forming a linear combination from each RGB light source, 4, 8 or 16 symbols can be created, encoding more than 1 bit in a symbol. CSK supports dimming by attenuating the current driving through LED.

2.2.4 OFDM

OOK and VPPM are binary modulation schemes. Orthogonal frequency division multiplexing (OFDM) and discrete multi tone (DMT) are common techniques in RF wireless communication and enable digital subscriber lines (DSL). OFDM is a spectrally efficient modulation scheme that decomposes a single-carrier into multiple subcarriers. Each subcarrier is orthogonal allowing data to be transmitted simultaneously on each subcarrier. Unlike RF wireless communication, which has in-phase and quad-phase channels, VLC has only one channel. Hence, OFDM architecture for RF communication cannot apply to VLC directly. One approach for adapting OFDM to VLC is applying Hermitian symmetry on the data to generate in-phase signal signal [6]. Although OFDM/DMT is not defined in the standard, it is still the most popular modulation scheme for high data rate VLC research.

Mode	Modulations	RLL code	Optical clock rate	FEC		Data rate
				Outer code	Inner code	
PHY I	OOK	Manchester	200 kHz	RS (15,7)	CC 1/4	11.67 Kbps
				RS (15,11)	CC 1/3	24.44 Kbps
				RS (15,11)	CC 2/3	48.89 Kbps
				RS (15,11)	None	73.3 Kbps
				None	None	100 Kbps
	VPPM	4B6B	400 kHz	RS (15,2)	None	35.56 Kbps
				RS (15,4)	None	71.11 Kbps
				RS (15,7)	None	124.4 Kbps
				None	None	266.6 Kbps
PHY II	VPPM	4B6B	3.75 MHz	RS (64,32)		1.25 Mbps
				RS (160,128)		2 Mbps
			7.5 MHz	RS (64,32)		2.5 Mbps
				RS (160,128)		4 Mbps
				None		5 Mbps
	OOK	8B10B	15 MHz	RS (64,32)		6 Mbps
				RS (160,128)		9.6 Mbps
			30 MHz	RS (64,32)		12 Mbps
				RS (160,128)		19.2 Mbps
			60 MHz	RS (64,32)		24 Mbps
				RS (160,128)		38.4 Mbps
			120 MHz	RS (64,32)		48 Mbps
				RS (160,128)		76.8 Mbps
				None		96 Mbps
PHY III	4-CSK	None	12 MHz	RS (64,32)		12 Mbps
	8-CSK			RS (64,32)		18 Mbps
	4-CSK		24 MHz	RS (64,32)		24 Mbps
	8-CSK			RS (64,32)		36 Mbps
	16-CSK			RS (64,32)		48 Mbps
	8-CSK			None		72 Mbps
	16-CSK			None		96 Mbps

Table 2.1: Summary of physical layer defined in IEEE 802.15.7-2001 standard [5].

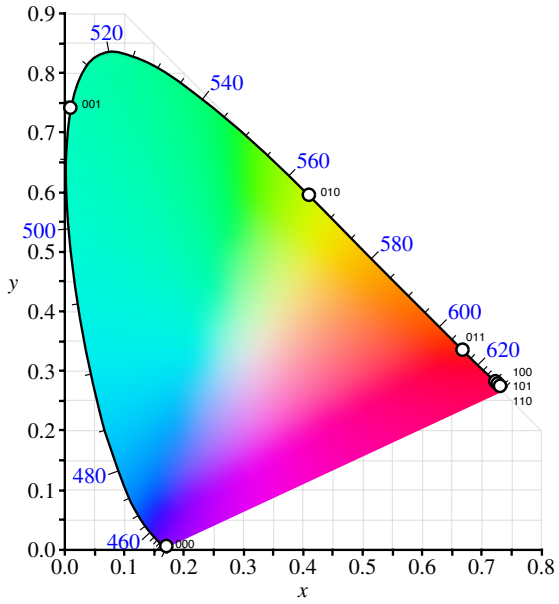


Figure 2.5: CIE 1931 color coordinates [40]. The 7 defined color bands are marked from 000 to 110. 3 bands out of 7 are chosen chosen to form vertexes of a triangle. By linear combination of 3 selected bands, 4, 8 or 16 symbols are created to achieve multiple bits per symbol.

2.3 VLC Applications

In this section, we visit related works that have been done on various different VLC applications. Broadly, much of this work focuses on high data rate communication and mobile applications.

2.3.1 High Data Rate Communication

The data rate of binary modulation is bounded by the impulse response. White color is generated by mixing RGB channels or synthesizing blue light with yellow phosphor. Most low-cost, non-tunable white LEDs are a phosphorescent type LED. However, the slow response time for yellow phosphor limits communication bandwidth. Figure 2.6 shows the impulse response of a phosphors type white LED. The steep rise corresponds to blue light while the long tail is the result of the yellow phosphor. One approach to suppresses the long tail from yellow phosphor to increases bandwidth is using blue optical filter at the receiver [36]. In addition, avalanche photodiode (APD) has shorter response time than PIN (p-type, intrinsic, n-type semiconductor) diode, so APD can be used to increase data rate. Researchers demonstrated a 230 Mbps optical link using OOK as a modulation scheme, and blue filter on APD type receiver [52].

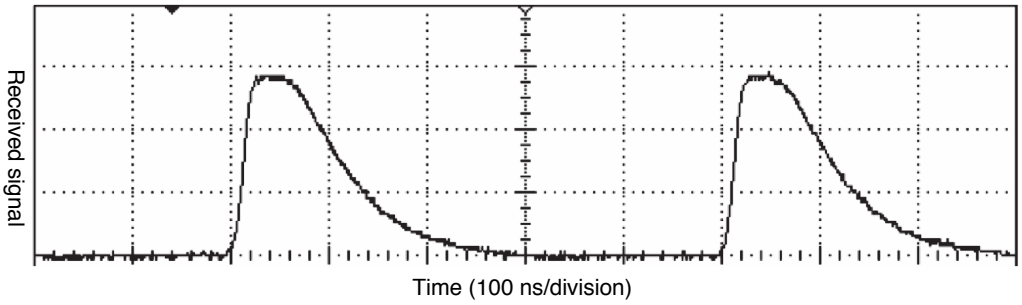


Figure 2.6: This figure shows the received impulse response without optical filter. The transmitter is generating an impulse response and receiver sees a sharp spike due to blue LED and a long tail comes from slow response time of yellow phosphor [47].

Figure 2.7: High speed VLC transceiver. Heinrich Hertz Institute developed a VLC transceiver operates at 500 Mbps at 4 m or 120 Mbps at 20m. It uses OFDM modulation to achieve high data rate and features bidirectional communication [35].

OFDM further pushes data rate. Grobe et al. demonstrated a bidirectional real-time VLC prototype which operates up to 500 Mbps [35]. Their system uses rate adaptive OFDM which changes data rate with fixed error rate. As mentioned earlier, colors are orthogonal channels that allow independent data to be simultaneously transmitted. Cossu et al. demonstrated an experimental setup using RGB colors to transmit data, and each color modulates data using OFDM. Authors demonstrates 3.4 Gbps bandwidth is possible [20].

2.3.2 Camera Communication

A CMOS image sensor can be used as a VLC receiver. However, limited frame rate limits the decode rate of a camera. Danakis et al. demonstrated a 3.1 Kbps data rate using a camera on mobile phone with 20 frames per second (FPS) by leveraging the rolling shutter

effect [24]. Screen-to-camera communication is a mobile phone to mobile phone communication link. One phone streams the information using the screen while the second phone uses its camera to receive the data. The communication links are fundamentally limited by inter-symbol-interference. The Casio Picapicamera mobile application uses flashing color on entire screen to encode data. Since the transmitter and receiver are not synchronized, Picapicamera has to flash at the rate lower than receiver's FPS to guarantee reception [15]. Furthermore, higher data rates can be achieved by using 2D colored barcodes, but motion blur reduces the robustness. COBRA adaptively adjusts data rate while sacrificing robustness while in motion [38]. Styrofoam introduces blank frames between transmissions to guarantee message decoding. The authors demonstrated a 69.1 kbps link using a 60 fps LCD screen as a transmitter and 30 fps smartphone camera as a receiver [59]. HiLight on the other hand, is not optimized for data rate. Instead it hides the information in a "static" video. Data is encoded in tiny translucency changes (alpha channel) that are imperceptible to humans [58].

2.3.3 Indoor Positioning/Navigation

Many common indoor positioning techniques are RF-based. These techniques include time-of-arrival (ToA), time-of-flight (ToF) or fingerprinting. However, these techniques have many drawbacks. They typically require dedicated infrastructure and significant RF bandwidth. Visible light on the other hand has short wavelength and large bandwidth, making it an attractive solution for indoor localization/navigation applications. Several visible light positioning systems have been implemented [57, 60, 73, 77]. ALTAIR uses ceiling-mounted cameras, body-worn IR LED tags, and a server that links tags to beacon sequentially, captures images from the cameras, and performs triangulation to estimate position [77]. Epsilon estimates location using received signal strength. Based on the line of sight (LoS) link model and the assumption of the receiver angle, it calculates a distance based on received signal strength (RSS) offering a result of half-meter accuracy. However, this model requires a custom light sensor that plugs into a smartphone's audio port as well as user calibration. [57]. Landmarks and Bytelight provide semantic (room-level) localization using CMOS camera on mobile devices [2, 73]. Landmark uses binary frequency shift keying (BFSK) to encode symbols, and camera captures multiple frames to decode the location ID. Similarly, Lim et al. uses a photodiode on shopping carts to receive location ID and further provide indoor navigation [60].

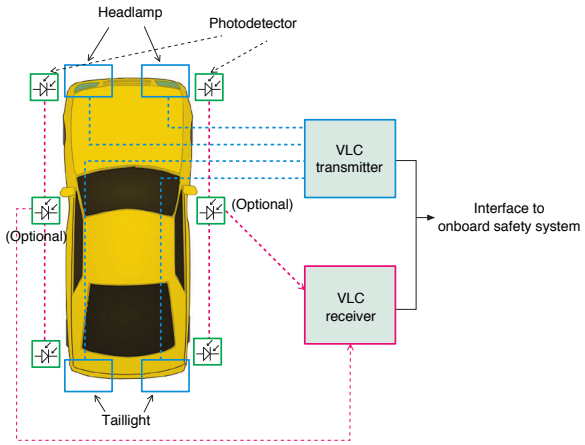


Figure 2.8: VLC on vehicle. The headlamp and taillight transmit data to adjacent vehicles, and surrounded photodectors receive from other vehicles. By sharing information, car with VLC technology could actively avoid collision [93]

.

2.3.4 Vehicle Communication

VLC has be used to enable vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication [13, 93]. Yu et al. demonstrated a low-cost and low-complexity VLC system on a vehicle that communicates with nearby line-of-sight vehicles. The headlamps and taillights on vehicle provide lighting, transmitting information to nearby vehicles. Using this method, vehicles could share their speed or calculate relative distance to avoid collision with automatic braking. V2V/V2I communication operates at outdoor environment. It therefore experiences a highly dynamic channel that changes with weather conditions such as fog, rain, snow and direct sunlight. Figure 2.8 illustrates the V2V communication.

2.3.5 Bi-direction Communication

When photons with enough energy strike an LED, current is produced, allowing an LED to be used as VLC receiver. A single LED can be used as a transmitter and a receiver by duty cycling LED [26]. Schmid et al. demonstrated a software-defined MAC layer for LED-to-LED VLC network [80]. The authors use LEDs as receivers within ”off“ period in idle pattern, and maintain time synchronization across multiple devices. They demonstrates 800 bps at 2 m using a simple, low-cost platform. OpenVLC is a software-defined VLC research platform [90]. It exports the VLC PHY and MAC layers to Linux kernel, so that applications could use it as a network interface. It provides certain flexibility to PHY layer and completely software-defined MAC layer. However, the OpenVLC is mainly targeting the low-cost, low rate communication applications.

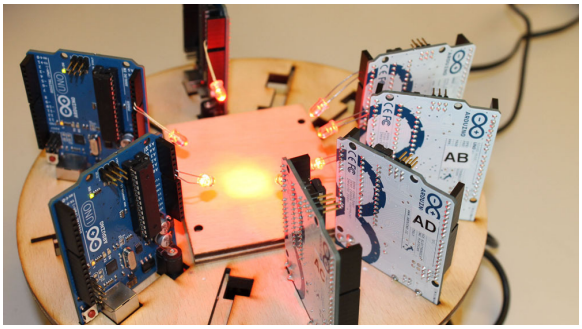


Figure 2.9: Bi-direction VLC. Schmid et al. demonstrate using LED for both transmitter and receiver. To avoid flicker and maintain data communication without a separated receiver, LED switches to RX mode every cycle during idle pattern, and a synchronization protocol keeps time between nodes [80].

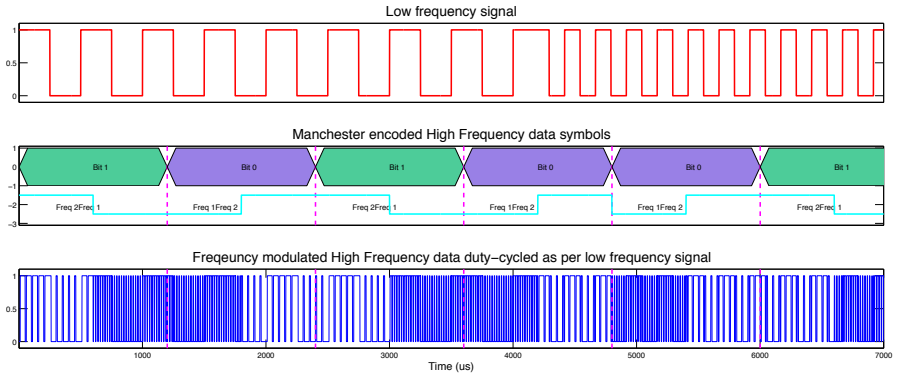


Figure 2.10: Hybrid VLC [72]. First row is data for low rate application. The "on" and "off" modulates the duty cycle of optical waveform. Second row is Manchester encoded FM signal for high rate application. The frequency applies to waveform directly.

2.4 Multiplexing Applications

The closest work to our vision of SDL is hybrid communication from researchers at CMU [72]. Rajagopal et al. demonstrated two VLC applications can be simultaneously realized by a single LED transmitter. The modulation waveform they used was a Manchester encoded frequency modulation (FM) with varying duty cycle. Manchester encoded FM uses the transition of two frequencies (f_1 and f_2) to represent symbols ($f_2 \rightarrow f_1 \Rightarrow$"1", $f_1 \rightarrow f_2 \Rightarrow$"0"). These encoded symbols are then captured and decoded by a camera receiver with an FFT. In the meantime, the duty cycle of the waveform is modulated by a low frequency signal (f_3, and $f_3 < f_1$, $f_3 < f_2$), which can be captured by a diffusing receiver with a low pass filter. The authors showed a low data rate link 1.3 Bps and a higher rate link 104 Bps can be achieved simultaneously.

CHAPTER 3

SDL Architecture and Implementation

In this chapter, we present the SDL system architecture, focusing primarily on the communication endpoints–the light transmitters and receivers. Transmitters consist of a control unit (e.g. microcontroller or FPGA), backhaul network link(s) (e.g. Ethernet, wireless, or power-line communication), and LED downlink. Receivers use custom photodiode circuits or mobile phone CMOS cameras to receive LED signals from the transmitters. We design several custom LED transmitters and photodiode receivers to evaluate the SDL architecture on. We further trivially modify several commercial LED luminaries into SDL transmitters and implement a receiver on a commercially available smartphone to demonstrate the real world viability of the SDL system and to explore a variety of applications.

3.1 Overall Architecture

While early work captures the landscape of possibilities for VLC [46, 69], newer works explore VLC throughput optimization [53, 88], additional applications for VLC [50, 73], and VLC integration with other technologies, such as power-line communication [45]. As solid-state lighting begins to mature and spread, the time is ripe to crystallize a flexible SDL architecture to springboard the vast possibilities of VLC off of.

To better enable researchers and other innovators to explore VLC-enabled networks, we propose the architecture shown in Figure 3.1. In the SDL architecture, each luminaire is outfitted with a control unit, which contains a data plane and a control plane. The data plane receives packets from the cloud or a cloudlet via a backhual network link(s), parses the packet, and passes on the relevant data to the control plane. The control plane converts data received from the data plane into an LED waveform, and emits the data via an LED frontend attached to the luminaire. In addition, the control plane is responsible for inter-luminaire communication, which is used to synchronize luminaire transmissions.

While the VLC component of the SDL architecture is independent of the data plane, we

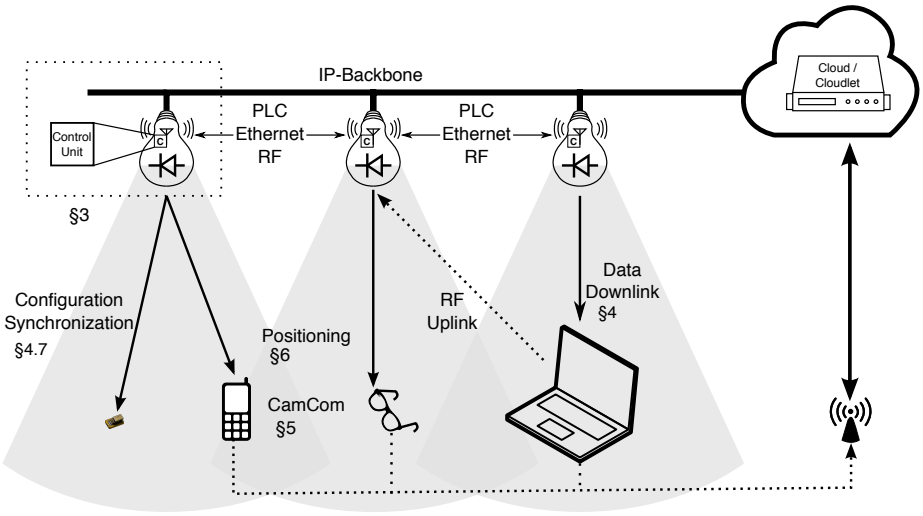


Figure 3.1: SDL architecture overview. Our architecture consists of networked smart luminaire transmitters along with a variety of receivers. Different types of receivers may be better suited to certain applications, such as localization and communication.

note that the choice of backhaul(s) for the data plane will have an impact on the capabilities of the overall SDL system. For example, the connectivity of an Ethernet or powerline network relies on physical network topology, whereas an RF-based approach relies on the geographic distribution of the lights and the nearby environment.

This architecture provides high flexibility for the research community, while also building off of decades of Internet technologies and networking research. Smart luminaires act as IP routers capable of participating in mesh routing protocols, and can potentially forward data across multiple network interfaces. This architecture also borrows concepts from software-defined networking (SDN), such as separating the control and data planes. From software-defined radios (SDR), this architecture borrows the idea of programmatic manipulation of physical layer waveforms. The SDL architecture focuses on flexibility, seeking to enable the widest class of applications while maximizing simplicity by leveraging existing infrastructure (IP) and experiences in modular networking design (SDN, SDL).

3.1.1 SDL Cloud/Cloudlet Data Structures and APIs

Each set of smart luminaires (e.g. the smart luminaires in a particular building) must be connected to and SDL server in the cloud or on a cloudlet. The SDL server serves as the interface between applications and the smart luminaires. The SDL server is responsible for relaying configuration information and data to the smart luminaires, and for providing access controls, system information, and task schedulability updates to applications.

3.1.1.1 Data Structures

The SDL server communicates to the smart luminaires using a variety of structured network packets. In addition to providing data to transmit, these data structures are used to provide configuration, state, and data encoding methods to the smart luminaires.

```
/*
Data Structure
*/
typedef enum luminaire_state{
        ON,
        OFF
} luminaire_state_t;

typedef enum luminaire_interface{
        ETHERNET,
        PLC,
        WIRELESS
} luminaire_interface_t;

typedef struct luminaire{
        uint32_t room;
        uint32_t id;
        uint8_t address[6];
        luminaire_state_t state;
        luminaire_interface_t interface;
} luminaire_t;

typedef struct luminaire_list{
        luminaire_t** luminaire_ptr;
        unsigned int numOfLuminaires;
} luminaire_list_t;

typedef enum encoding_type{
        OOK,
        MANCHESTER
} encoding_type_t;

typedef struct sdl_data_packet{
        uint8_t* data_ptr;
        uint8_t data_length;
        uint8_t symbol_dutyCycle;
        uint32_t symbol_rate;
        encoding_type_t data_encoding;
        uint8_t data_priority;
        uint8_t repeatInterval;
        uint8_t numOfRepeat;
        uint32_t timeToTransmit;
} sdl_data_packet_t;
```

```
typedef struct sdl_cfg_packet{
        luminaire_state_t luminaire_state;
        uint8_t idlePattern_freq;
        uint32_t idlePattern_dutyCycle;
} sdl_cfg_packet_t;

typedef struct super_frame{
        uint8_t numOfSlots;
        uint8_t slotFrequency;
}frame_t;

typedef enum schedule{
        SCHEDULED_SUCCESS,
        SCHEDULED_CONFLICT,
        SCHEDULED_FAILED
} scheduled_t;
```

3.1.1.2 APIs

The SDL server APIs allow applications to interface with smart luminaires while providing administrators with a centralized access control system. These APIs provide smart luminaire locations and availability, and allow applications to send data to and reconfigure smart luminaires. The SDL administrator is responsible for deciding which APIs specific applications or application groups should have access to.

```
/*
List all available luminaires
Return a list of available luminaires
*/
luminaire_list_t db_list_all_luminaires();
```

```
/*
List luminaires from specific room
Return a list of available luminaires
*/
luminaire_list_t db_get_luminaires_from_room(unsigned int room);
```

```
/*
Get VLCP frequency configuration
Return frequency of specific luminaire
*/
unsigned int db_get_vlcp_freq(unsigned int room, unsigned int id);
```

```
/*
Transmit data packet to LED luminaire
Input: data packet, destination address
Output: scheduled status
*/
scheduled_t sdl_data_packet_transmit(sdl_data_packet_t dataPkt, char* address);
```

```
/*
Configure LED luminaire
Input: configuration packet, destination address
*/
void sdl_cfg_packet_transmit(sdl_cfg_packet_t cfgPkt, char* address);
```

```
/*
Initialize Data plane
*/
void data_plane_init();
```

```
/*
Configure super frame
*/
void superFrame_config(frame_t frame_cfg);
```

```
/*
Determines if transmitting a piece of data
will result in a visible flicker in the LED luminaire.
Input: Data packet, configuration packet
Output: If data causes potential flicker
*/
bool data_flicker_examine(sdl_data_packet_t dataPkt, sdl_cfg_packet_t cfgPkt);
```

3.2 Smart Luminaires

Smart luminaires are application controllable LED luminaires, providing a bridge between an IP backhaul and optical receivers. By allowing applications to control LED state, smart luminaires help to enable a wide variety of post-illumination applications, such as localization, communication, and entertainment. However these functions cannot impede the main purpose of LED lighting—illumination. The requirements of illumination differ from location to location. For example, ideal color temperature and light intensity varies between different settings (e.g home and office) to increase mental health and productivity [62, 66].

Thus, smart luminaires must be flexible enough to support a wide variety of locations, each with its own resources and requirements. In addition to having different illumination requirements, different locations may have different available IP backhauls. These varying resources and requirements leads us to modularize the SDL smart luminaire into three

major components, the *LED frontend*, the *control plane*, and the *data plane*. The LED frontend is composed of a high intensity LED or LED array and other required electronics to drive the LED. The main purpose is providing illumination and optical communication. The control plane controls the LED frontend and the data plane, and provides time synchronization between smart luminaires. The data plane provides a network backhaul from the smart luminaire to the SDL server. The particular backhaul it provides (e.g Ethernet, PLC, Wi-Fi) can vary with environment. The data plane is responsible for parsing incoming IP packets, thus providing the control plane with a standard interface.

3.2.1 Design Spaces

In order to implement an effective SDL system, we must first understand the trade-offs, constraints, and requirements of the LED frontend, the control plane, and the data plane.

3.2.1.1 LED Frontend

The major constraint of the LED frontend is *required bandwidth*, which is application dependent. Common modulation schemes for VLC such as binary modulation (ex. OOK, VPPM) and multi-level modulation (ex. OFDM) support different bandwidths and require different hardware. Binary modulation is simple to implement, but provides low bandwidth. In contrast, multi-level modulation schemes offer high communication rates, but increase system complexity for both transmitters and receivers.

The commercial LED bulbs can be trivially instrumented to transmit data using binary modulation at ˜1 Mbps. However, commercial LED bulbs have a wide range of configurations, which may require different control units. Most commercial LED bulbs (ex. Commercial Electric's T66) use a hybrid in series and in parallel LED array configuration, so that it has a forward voltage drop of around 20-40 V with a ˜500 mA current draw. In contrast, a few commercial LED bulbs (ex. Commercial Electric's T67) use a pure in series configuration for their LED arrays, resulting in a higher forward voltage drop (˜320 V).

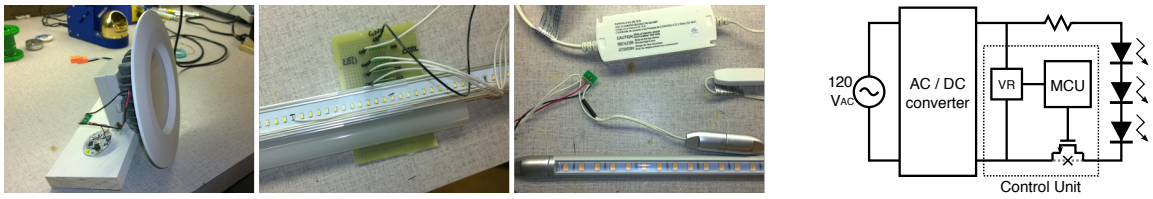
In addition to powering the LED Frontend from an AC/DC converter, it is also possible to supply power via Power over Ethernet (PoE). PoE offers up to 25.5 W [4], which is sufficient for common LED bulbs that consume ˜15 W (80 W incandescent equivalent). However, commercial LED bulbs have different I-V curves and operation points, so powering from PoE may require additional circuits to limit LED current draw when a higher supply voltage is applied.

3.2.1.2 Control Plane

The control plane is responsible for processing data from the data plane, generating optical waveforms for the LED frontend, and synchronizing smart luminaires. Synchronicity between smart luminaires is measured as absolute time jitter divided by optical symbol rate. Low-cost MCUs and FPGAs can both be used to power the control plane, each with its own trade-offs. Low-cost MCUs provide a simple, compact, and integrated control plane, but offer lower optical data rates and loose synchronicity. In contrast, FPGAs have more deterministic timing which allows for tighter synchronicity and higher optical data rate applications, but carry higher development costs. Combining MCUs and FPGAs increases the SDL application domain, but providing the right interface between the MCU and FPGA can be challenging.

3.2.1.3 Data Plane

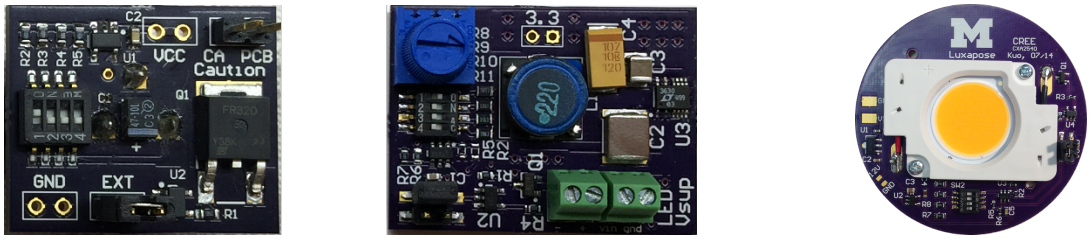
The data plane provides an IP backhual to the smart luminaire. The data plane's main design constraints are available bandwidth, potential applications, and infrastructure support. The obvious candidate backhauls are power line communication (PLC), Ethernet, and Wi-Fi. PLC and Ethernet offer considerable bandwidth, which is sufficient for instrumented commercial LED luminaires with simple binary modulation schemes, and both of them provide data and power in a single connection. However, deploying Ethernet to each smart luminaire requires significant infrastructure changes in existing buildings. PLC modulates data onto existing AC wiring, leveraging infrastructure already required for lighting. However, appliances switching on and off and switching power supplies, which can be found in many AC/DC converters, can produce glitches on the AC main [87], producing a noisy communication environment. In addition to noise, commercial buildings often have separate wiring domains isolated by transformers that imposes high impedance on high frequency domains. As a result, PLC cannot provide connectivity across different wiring domains. Wi-Fi is an attractive option due to its prevalence, but may require significant upgrades to a building's Wi-Fi infrastructure, and raises the hardware cost of the data plane.



(a) LED light fixtures

(b) Custom control unit.

Figure 3.2: COTS light fixtures and control unit. Augmenting the COTS fixtures requires only inserting a single FET in their power supply controlled by an external microcontroller.



(a) Control unit for commercial LED luminaires.

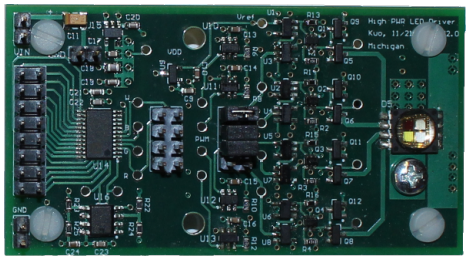
(b) Controller w/ LED.

Figure 3.3: Figure 3.3a instruments commercial LEDs to transmit data. The left board is designed for a high input supply voltage (˜400 V) without isolation, whereas the right board is designed for a 60 V supply voltage. Figure 3.3b integrates a CREE CXA2540 LED. The dip switch on the boards configures pure tone frequency that is being used in a VLCP application.

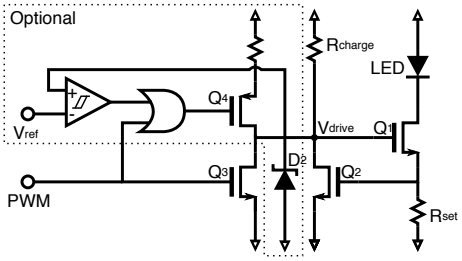
3.2.2 Exploring Design Spaces

We have successfully modified many commercial LED luminaires, including CAN lights, tubes, and task lamps. Most commercial LED luminaires can be trivially modified into an LED frontend by cutting a wire, adding a transistor, a voltage regulator, and an oscillator or MCU, which is shown in Figure 3.2. Furthermore, we built a custom LED frontend which integrates a current limiting circuit and feedback loop to enhance bandwidth. We also designed several compact systems that enable commercial LEDs to transmit location beacons for positioning applications. Figure 3.3a shows the LED frontend as a commercial LED add-on and Figure 3.3b shows our custom integrated version. These LED frontends contain a resistor bank which configures an oscillator to set location beacon frequencies for VLC positioning applications.

In addition, we have designed a RGB LED transmitter which allows data to be encoded using orthogonal channels. It can used for high data rate applications. However, if the data modulation at each of the three channels (RGB) is independent, the overall color will be data dependent. In a human friendly lighting modulation scheme, the data modulation at



(a) RGB LED transmitter.



(b) Current Limiter.

Figure 3.4: Multi-channel transmitter. All channels are independent with current limiting circuit. With optional transistors, the LED transmitter is capable of transmitting data in ˜20 Mbps.

each channel cannot be completely independent. Pre-processing is required to guarantee consistent color. Figure 3.4 shows the multi-channel transmitter and the schematic of our current limiter and feedback loop. The right half of the RBG transmitter is a simple current limiting circuit. The current flow through each LED is equivalent to the current flow through R_{set}. The voltage drop across R_{set} is equivalent to the V_{gs} of NFET Q_2. R_{set} is chosen to set the current at the steady state. If current is larger than the steady state, the overdrive voltage of Q_2 also increases, which reduces the voltage of V_{drive} and V_{gs} of Q_1, causing current draw to decrease. NFET Q_3 functions as a switch to discharge the V_{drive} to turn off the LED. Once the PWM is de-asserted, the circuit will reach the steady state by charging through R_{charge}. A smaller R_{charge} has a faster charging rate, but consumes more power when the LED is off. Hence, we designed additional logic which triggers PFET Q_4 to provide a fast charging path. This path is disabled while the LED is off. The additional charging circuit significantly increase the bandwidth, while maintaining low power consumption at R_{charge}.

To explore the design space of the control plane and data plane, we have designed a flexible platform that includes many peripherals in an integrated design. Figure 3.5 shows our smart luminaire control+data plane experimental platform. This platform contains a Microsemi SmartFusion, which is a highly integrated hardcore ARM Cortex M3 connected to a flash based FPGA and analog/digital converters. In addition, this platform includes a software-defined radio (SDR) that is composed of high speed ADC/DACs and a 2.4 GHz RF transceiver. The SDR enables novel wireless research to be integrated into the SDL system. This can be used to provide services such as wireless flooding or time synchronization the the smart luminaires. Our platform provides a Ethernet IP backhaul for the data plane, although other backhauls can be integrated via headers.

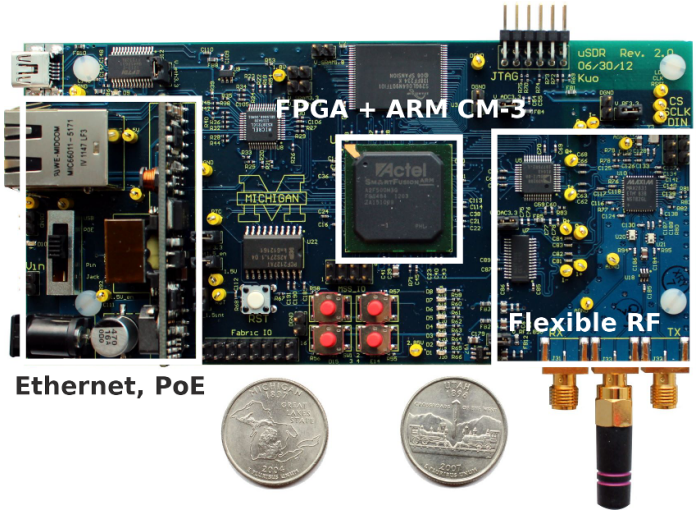
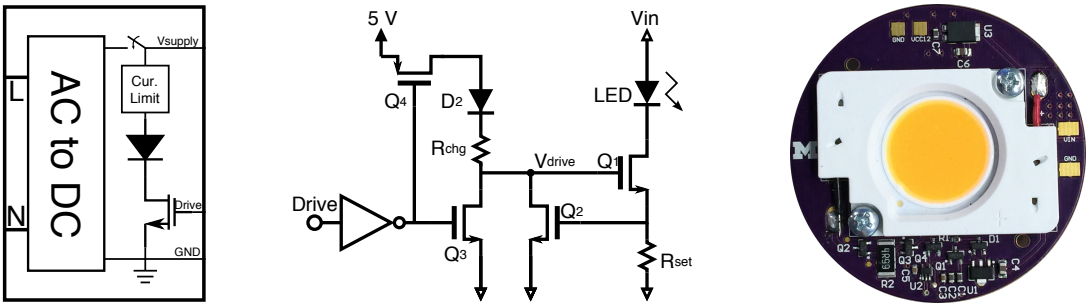


Figure 3.5: Integrated control plane and data plane. This platform integrates control plane and two data plane options (Ethernet, and wireless radio).

3.2.3 Design Decisions and Implementation

We evaluate possible implementations in terms of *cost* and *deployability*. A widely deployable system must be low cost, small, and flexible. The cost of the system includes both hardware and labor costs. Ideally, our system would have a low hardware cost and a low installation cost. Furthermore, the SDL implementation should support a variety of services and applications while requiring minimal hardware changes. Deployability refers to the ease in which users or building administrators could actually deploy an SDL implementation. The main factor influencing deployability is how much additional infrastructure an implementation would require.

VLC has rich applications in illumination, entertainment, communication, localization, device configuration, and health. We can categorize these applications into ultra high ($>$100 MBps), high(100$\geq$ 1 Mbps), and low data rate ($<$ 1 Mbps) applications. While supporting all data rates would be ideal, in practice there is a trade-off between data rate, cost, and deployability. Ultra high data rate VLC commonly uses the OFDM modulation scheme, which modulates data using light intensity. OFDM can provide 300 Mbps, but can significantly raise hardware and labor costs while lowering deployability. Due to the non-linear I-V curves of LEDs, an OFDM transmitter would need to be calibrated for each LED luminaire [68], raising labor costs. In addition, OFDM transmitters require custom luminaires and more complicated LED frontends compared to on-off keying transmitters. On the receive side, OFDM requires custom photodiode receivers with complex signal processing to demodulate the transmitted signal, precluding smartphone applications and raising receiver hardware costs. Simple on-off keying can support high data rate VLC ap-



(a) LED arch. (b) Schematic of current limiter. (c) LED implementation.

Figure 3.6: LED Frontend schematic and implementation. Figure 3.6a shows the functional diagram. LED Frontend can be supplied by an AC/DC converter or PoE. To prevent sinking too much current from PoE and maintain constant brightness, a current limiting circuit is required. The schematic is shown in Figure 3.6b.

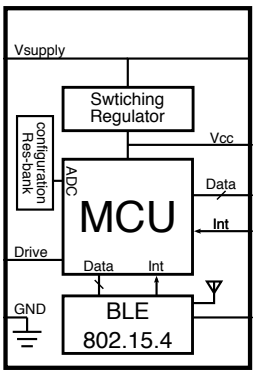
plications, but may require custom luminaires. However, commercial luminaires with a few trivial modifications can support low data rate applications. While many potential applications may require ultra high or high data rates, we optimize our implementation for low data rate applications. This is because mobile devices can currently only act as low speed receivers, and we argue that *mobile devices are the key enabler for VLC technology.* Mobile devices such as smartphones and tablets provide a ubiquitous platform to serve as SDL receivers. We believe that targeting mobile devices is much more fruitful than attempting to deploy custom LED receivers.

3.2.4 Smart Luminaire Implementation

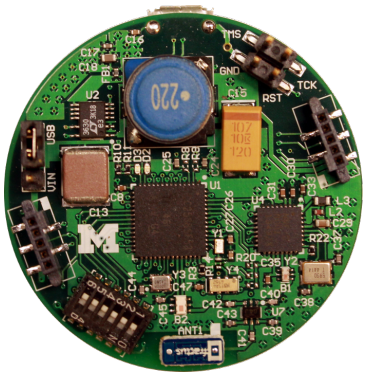
In addition to our experimental platform, we implemented a integrated smart luminaire suitable for actual deployments. This smart luminaire can either be used with a custom LED frontend or with a modified commercial luminaire, and is optimized for low data rate communication with mobile phones.

3.2.4.1 LED Frontend

We designed a simple LED frontend that includes a high brightness LED, a current limiting circuit, and a control feedback control loop. The LED frontend can be powered by an AC/DC converter directly or from a PoE interface. The circuit is configured to provide the LED with a fixed current draw over a wide range of supply voltages. Figure 3.6b shows a schematic of our implementation of the current limiter and feedback control loop for the CREE CXA2540 LED. Q_4, D_2, and R_{chg} provide a fast, low-power, and simple charging



(a) Control Plane.



(b) Implementation of Control Plane.

Figure 3.7: Figure 3.7a shows the functional diagram of Control Plane. It is composed of a power supply, a microcontroller, a low-power radio and a BLE radio. Figure 3.7 is the implementation with stackable connectors and configuration dip switch.

path to reduce LED turn on time. R_{set} fixes the current draw and voltage drop of the LED for a 36-60 V V_{in}.

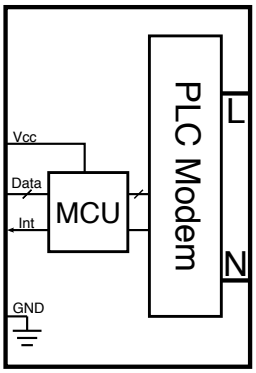
3.2.4.2 Control Plane

The control plane is the main processing unit which controls all peripherals, including the LED frontend and data plane. Moreover, the control plane has to provide low-voltage DC supply voltages to the rest of the smart luminaire. To achieve highly-efficient voltage regulation under a wide input voltage, a switching regulator is preferred. In addition to a microcontroller, a low-power wireless radio (IEEE 802.15.4) and a Bluetooth low energy (BLE) radio can be used to receive control messages, wireless synchronization, and to transmit metadata such as smart luminaire coordinates. Our current implementation uses a TI CC2538 SoC, which is a ARM-cortex M3 MCU integrated with a IEEE 802.15.4 radio, a Nordic NRF8001 BLE transceiver, and a Linear Technology LTC3630 high efficiency step down converter that provides 3.3 V to the smart luminaire.

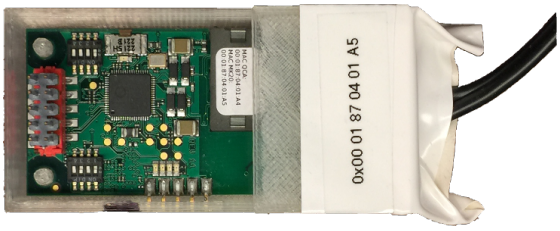
3.2.4.3 Data Plane

The data plane provides an IP backhaul to the smart luminaire. In our experimental platform, we provide built in PoE/Ethernet along with an optional PLC daughterboard. The Ethernet data plane is shown in Figure 3.5

We didn't see an option is superior than others in our application domain. Hence, we implemented both Ethernet and PLC data plane. Ethernet data plane is shown in our experimental platform (Figure 3.5). We use an I2SE PLC modem for our PLC data plane [39].



(a) PLC Data Plane



(b) I2SE Stamp 1 PLC development kit.

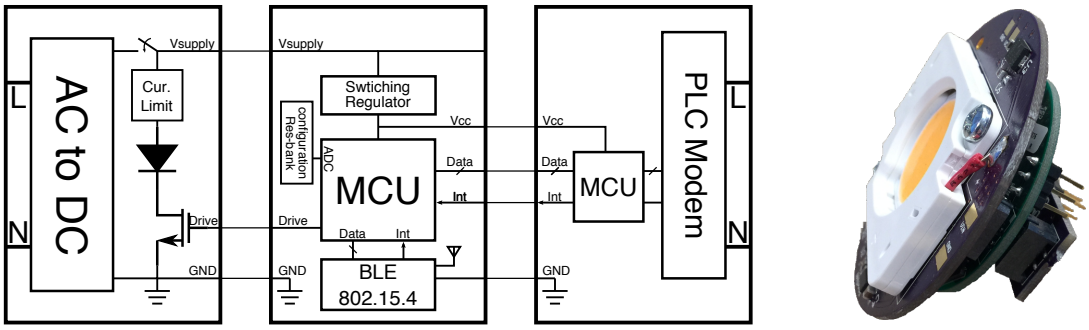
Figure 3.8: PLC Data Plane and commercial platform. Figure 3.8a shows the architecture of our PLC data plane, and Figure 3.8b shows the I2SE Stamp 1 dev kit we use for our PLC data plane. The I2SE Stamp 1 contains a QualComm QCA7000 (PLC Modem) and a FreeScale MK20D (external MCU).

MODULE	PART NUMBER	DESCRIPTION
LED Frontend	CREE CXA2540	High flux, multi-die integrated LED array
Control Plane	TI CC2538	SoC, IEEE 802.15.4 radio and Cortex-M3 MCU
	Nordic nRF8001	BLE connectivity
	Linear Tech LTC3630	High efficiency step down converter
Data Plane	Micrel KSZ8721CL	Ethernet PHY Layer Transceiver
	Freescale MK20D	PLC host controller, Cortex-M4 MCU
	QualComm QCA7000	IEEE 1901, HomePlug PLC modem

Table 3.1: Major components on our modularized smart luminaire implementation.

3.2.5 Integrated Smart Luminaire

Figure 3.9a provides a broad overview of our smart luminaire implementation. In our implementation, the control plane provides 3.3 V supply to all subsystem from input voltages ranging from 34 V to 55 V. The control plane exports 3 independent control channels to the LED frontend, along with a SPI interface and interrupt input pin to the data plane. The data plane uses the interrupt pin to alert the control plane of available data, which the control plane retrieves using the SPI interface. Figure 3.9b and Table 3.1 shows our implementation and a table of major components.



(a) Architecture of modularized smart luminaire.

(b) Implementation.

Figure 3.9: Integrating LED frontend, Control Plane and Data Plane. Figure 3.9a shows the interconnects between the three subsystems using PLC as the data plane. Figure 3.9b , although the PLC Data Plane is not shown.

3.2.6 Control Plane APIs

This section outlines the low-level software APIs to control the LED frontend. The proposed API uses a linked list to store out-going VLC packets. Each VLC packet includes *number of repetitions*, *repeat interval*, *time to transmit*, and *packet info*. Packet information contains specific parameters such as encoding schemes, symbol rate, duty cycle, payload length and data payload. The control plane application uses the *insert nodes* function to insert a packet into link list, which is sorted based on *time to transmit*. Once a packet is ready to be transmitted, it will be uploaded to a FIFO. The control plane automatically converts data in the FIFO into a PWM signal, which modulates the LED frontend and converted into an optical waveform. The *numOfRepeat* field is automatically decremented each time a packet is serviced, and packets are automatically removed when *numOfRepeat* is equal to 0.

```
/*
Data Structure
*/
typedef struct sdl_packet_node{
        sdl_data_packet_t* packet;
        uint8_t repeatInterval;
        uint8_t numOfRepeat;
        uint32_t timeToTransmit;
        struct sdl_packet_node* nextNode;
} sdl_packet_node_t;
```

```
/*
Querry transmitter.
Return true if LED luminaire is actively transmitting data.
*/
bool isTXBusy();
```

```
/*
Enable or disable LED luminaire.
*/
void output_enable(bool en);
```

```
/*
Set LED polarity (Active high/low)
*/
void set_output_polarity(bool pol);
```

```
/*
Return available visible light encoding scheme.
*/
outputEncodingScheme_t availableEncodingScheme();
```

```
/*
Set idle pattern.
Input: Duty cycle, frequency
*/
void setIdlePattern(float dutyCycle, int freq);
```

```
/*
Load VLC packet into TX fifo.
Input: A pointer to VLC packet.
*/
void pkt_load(vlc_packet_t* pkt_ptr);
```

```
/*
Parse configuration packet
Input: buffer from Data plane, pointer to configuration packet
*/
void parse_cfg_packet(uint8_t* buffer, sdl_cfg_packet_t*);
```

```
/*
Parse data packet
Input: buffer from Data plane, pointer to data packet node
*/
void parse_cfg_packet(uint8_t* buffer, sdl_packet_node_t*);
```

```
/*
Check linked list if outgoing packet should be loaded into TX Fifo
*/
void check_list_from_head();
```

```
/*
Insert packet to linked list
Input: Pointer to a data packet
*/
void insert_nodes(sdl_packet_node_t* );
```

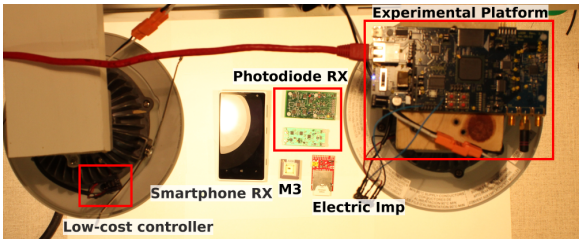


Figure 3.10: SDL components. Two commercial LED lights are controlled by two controller prototypes, a high-cost experimental platform and a low-cost fixed-function controller, respectively. The lighting data is being received by several receivers: 2 custom photodiode receivers, a M^3 "smart dust" [54] receiver, a Electric Imp Receiver [31], and a smartphone. Except for the smartphone, each receiver is a diffusing receiver.

3.3 Receivers

VLC receivers commonly use photo detectors such as photoresistors, phototransistors, photodiodes, and cameras. These work based on the photoelectric effect, which is that photons with enough energy will produce free carrier electrons (photocurrent) when they strike a detector material. By detecting changes in photocurrent produced by photo detectors, receivers can reconstruct light patterns.

Photoresistors change resistance based on light intensity, and can be found in many smartphones, tablets, and laptops, where they are used to adjust screen brightness. A phototransistor is a transistor with photo detector junctions. Light generates photocurrent at the base-collector junction, which is amplified by the transistor current gain. Similarly, a photodiode also produces photocurrent, but with a much faster response time than a phototransistor. For this reason, photodiodes are the most common photo detectors used in VLC applications, since bandwidth is directly proportional to response time. However, the most popular photo detectors are digital cameras, which can be found on the vast majority of smartphones. Digital cameras found on smartphones are typically CMOS/CCD cameras, which utilize an array of tiny photodiodes with RGB color filters. CMOS cameras receive data at a higher rate than their supported frames per second (FPS). VLC receivers can be categorized into two main types: diffusing and camera receivers. These receivers differ in if they use single cell or array photo detectors, and out of focus versus focused optical lens. Figure 3.10 shows a collection of SDL components, including both diffusing receivers (custom photodiode receivers, M^3, and the Electric Imp) and a camera receiver (smartphone). We will further delve into the challenges and limitations with communicating with diffusing and camera receivers in §4 and §5.

CHAPTER 4

VLC to a Diffusing Receiver

In this chapter, we explore providing microsecond-level synchronized transmissions from smart luminaires, which is useful for communicating with diffusing receivers. Diffusing receivers use an out-of-focus lens, meaning incoming light is mixed before the receiver sees it. Thus, visible smart luminaires must either be synchronized to avoid colliding transmissions, or use frequency data modulation. However, it is unrealistic to expect energy-constrained receivers, such as "smart dust" sensors, to demodulate frequency modulated data. Thus, we focus on providing microsecond-level time synchronization between neighboring smart luminaires.

Our goal is to allow smart luminaires in a room to transmit identical messages in "unison", maximizing the received signal strength for diffusing receivers in that room.

4.1 Receiver Architecture

Figure 4.1 shows the block diagram of a diffusing VLC receiver. The receiver is composed of photodiodes, amplifiers, interface logic, and digital back-end processing. The

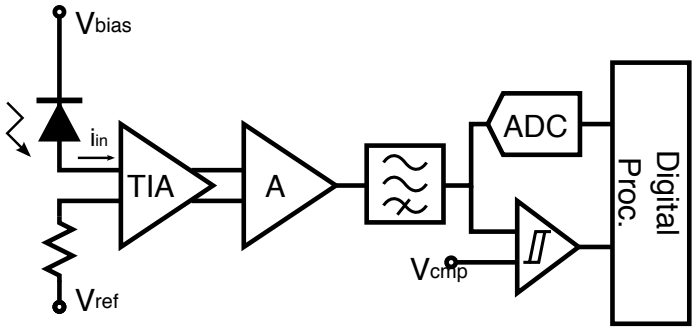
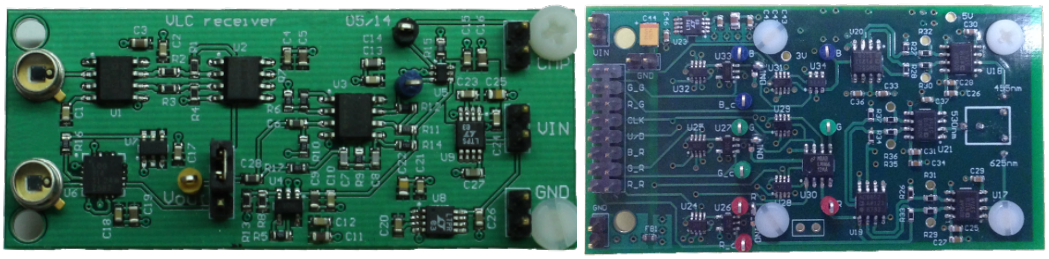


Figure 4.1: Diffusing receiver architecture. A receiver is composed of photodiodes, a transimpedance amplifier (TIA) that converts photocurrent to voltage, additional gain stage(s), a high/band pass filter, converters, and a digital processing module.



(a) Dual VLC receiver.

(b) RGB Linear receiver.

Figure 4.2: Prototypes of diffused VLC receivers. Figure 4.2a is a dual receiver. The two channels connect to two types of TIA (Log and Linear) that allow us to explore the trade-offs for the architecture. Figure 4.2b shows a multi-channel linear receiver. The three photodiodes have different optical filters that select RGB to pass through. The multi-channel receiver could enhance data rate by sending data using three channels simultaneously.

photodiode converts light into photocurrent, and its wide field of view (FoV) lens allows it to collect more photons. Some photodiodes contain optical filters. Photodiodes with highly selective optical filters could be used for wavelength multiplexing or receiving data from color encoded signals (eg. CSK). A photodiode can be operated in two regions—photovoltaic and photoconductive. A photodiode is operated at 'photovoltaic" mode if zero bias voltage applied, In this mode, dark current is reduced, but the photodiode experiences slower switching speeds due to higher junction capacitance. On the other hand, a photodiode is in "photoconductive" mode when it is reverse biased. In this mode, junction capacitance is smaller, resulting in higher switching speeds and higher dark current. Photoconductive mode is more common in communication applications due to its higher switching speeds, and thus higher bandwidth. The photocurrent from the photodiode is converted into voltage using a transimpedance amplifier (TIA) for easy manipulation (eg. amplifying, filtering). Dynamic range and bandwidth are fundamental trade-offs when selecting a TIA. Wide dynamic range allows larger input power variation due to changes in light intensity. A high pass or band pass filter is often required to remove the low-frequency interference from direct sunlight (DC), fluorescent lights (120 Hz), and other out-of-band signals. However, if the photodiode and ADC are not saturated, filtering can be done purely in the digital domain. The interface to the digital back-end could be a simple comparator for binary modulated signal (OOK, VPPM), or an ADC for multi-level modulated signal (CSK, OFDM). Figure 4.2 shows our prototype implementation. The dual receiver explores the trade-offs between input dynamic range and bandwidth using a linear and a log type of TIA. The RGB linear receiver is able to decode data from several color channels simultaneously, increasing communication bandwidth.

4.2 Synchronous Lighting

Unlike camera receivers, the optical lens of a diffusing receiver is designed to aggregate as much light as possible. Hence, a photodiode often has a large viewing angle, causing multiple light sources to mix at the receiver. Because of this, non-synchronized smart luminaires can result in colliding transmissions at diffusing receivers.

To avoid collisions at diffusing receivers, common multiple access techniques, include time division multiple access (TDMA), code division multiple access (CDMA) and frequency division multiple access (FDMA) may be used. However, pure TDMA does not scale well since only one light is transmitting at a time, so larger number of transmitters will result in flickering. FDMA in visible light communication has two interpretations. 1) *frequency of light*, and 2) *frequency of data*. Frequency of light uses different colors to transmit data, and this technique has been proposed to increase the data rate. [20] However, this technique heavily relies on the optical filter on the receiver. Unlike the FDMA on radio communication, the frequency selectivity of optical filters is usually poor and unadjustable. Therefore, this approach has limits the number of transmitter receiver pairs, causing poor scalability. Frequency of data transmission uses different symbol rates. The frequency selectivity of this approach imposes a huge computation requirement on receivers. This frequency selectivity has to be implemented in the digital domain, and requires a high sampling rate and a complex FIR filter. CDMA uses orthogonal codes to modulate data, requiring tight time synchronization [86]. Moreover, CDMA encounters the *near-far problem*. O One solution is to dynamically adjust the transmit power of all transmitters [30, 71]. However, this approach requires a feedback loop from receivers to transmitters, and adjusting output power would result in a noticeable effect on lighting.

Instead, we implement on-off keyed synchronized transmissions. That is, all transmitters in a given area send the same signals simultaneously. By doing this, the optical signal constructively interferes at the receiver, increasing receive power. Furthermore, this approach keeps receiver complexity and power usage to a minimum.

4.3 Increasing VLC Range

One advantage of synchronized transmissions is to potentially increase our communication distance. We envision a scenario in which a room has multiple sensors equipped with SDL receivers that require identical data, such as a firmware upgrade. We assume that a node has to receive a minimum optical power to correctly decode a packet. From the line-of-sight

(LoS) optical channel we discussed in §2.1, link loss (L_L) can be expressed as:

$$L_L \approx \frac{A_r(m+1)}{2\pi D^2} cos^m(\beta) \times cos(\alpha)$$

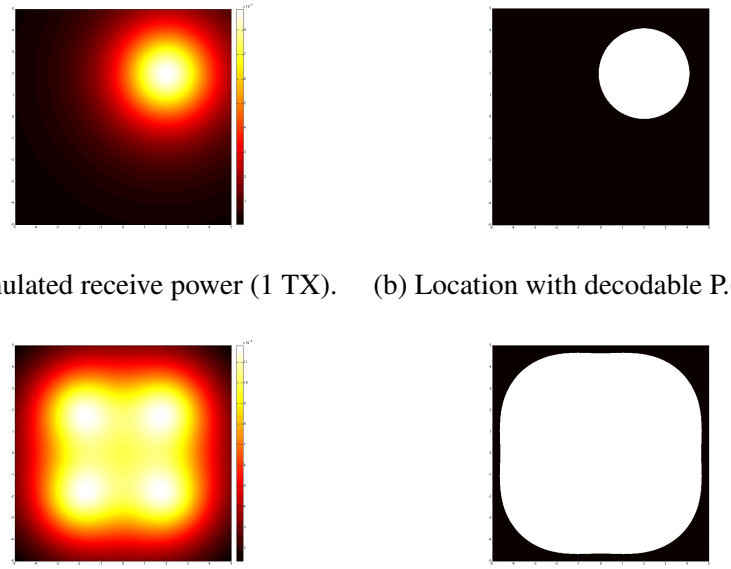
Assuming the order of Lambertian emission (m) is 1 and angle $\alpha = \beta = \theta$, optical path loss can be further simplified to $\frac{A_r}{\pi D^2} \times cos^2(\theta)$. A_r is the photo detectors area, D is the distance between the transmitter and receiver, and θ is the angle between the normal direction of the transmitter (or receiver) and the direct connection between the transmitter and receiver. If the receiver is located at $(x_r, y_r, 0)$ and transmitter i is located at (x_i, y_i, z_i), the angle θ can be written as: $tan^{-1}(\sqrt{(x_i-x_r)^2+(y_i-y_r)^2}/_{z_i})$ Also, $D = \frac{z_i}{cos(\theta)}$. Hence, given the transmitter's location, optical path loss at any location can be expressed as:

$$L_{L_i} = \frac{A_r}{\pi \times z_i^2} \times cos^4(tan^{-1}(\frac{\sqrt{(x_i - x_r)^2 + (y_i - y_r)^2}}{z_i}))$$

The received optical power at any wavelength can be calculated using super position of transmitted power at given wavelength and path loss.

$$S_r(\lambda) = \sum_{i=1}^{n} S_{t_i}(\lambda) \times L_{L_i}$$

We simulated the receive power at a 0.5 cm×0.5 cm photo detector with single and multiple transmitters in a 10 m×10 m×3 m space. The transmitters are placed 3 m ($z_i = 3$) above the receiver, which is located on the floor of our simulated space. The transmitters face directly down, while the receiver faces directly up. We assume that the path loss has to be less than 4×10^{-7} for the receiver to correctly decode the symbol. First, we placed a transmitter at location $(2, 2, 3)$ to see the coverage (locations with less than 4×10^{-7} path loss) of a single transmitter. Figure 4.3a shows the coverage on the floor ($z = 0$). The white area indicates locations that receive sufficient optical energy. We found that a single transmitter is able to cover 13.73% of the floor. Hence, with optimal light placement and time multiplexing transmitters, we require at least 8 transmitter to cover the entire floor. On the other hand, synchronized transmitters can increase VLC communication range by superposition. Figure 4.3c shows the heat map of 4 transmitters. This simulation shows that 4 transmitters with the same TX power as our previous simulation cover 78.02% of the floor when utilizing synchronized transmissions.



(a) Simulated receive power (1 TX). (b) Location with decodable P.(1 TX).

(c) Simulated receive power (4 TXs). (d) Location with decodable P. (4 TXs).

Figure 4.3: Simulated receive power. In Figure 4.3a and Figure 4.3c, brighter areas represent stronger received signal, and white areas in Figure 4.3b and Figure 4.3d represent areas with sufficient receive power to decode a VLC signal. The 4 transmitter simulations assume the 4 transmitters are synchronized.

4.4 Multihop Smart Luminaire Synchronization

In order to reap the increased communication range of synchronized transmissions, we must first provide time synchronization to our smart luminaires. One way to do this is to utilize the Precision Time Protocol (IEEE 1588), which provides sub micro-second time synchronization over Ethernet. However, this requires Ethernet to be deployed to each smart luminaire, which may be prohibitively costly.

Hence, we design **Floodcasting**, a sub-microsecond wireless time synchronization protocol. In addition to time synchronization, Floodcasting also provides an efficient and reliable method of broadcasting a large volume of data to a network of wireless nodes.

In a Floodcasting network, an initiator starts a a Floodcast by streaming data into the network. Nearby nodes hear this Floodcast, and proceed to broadcast this data. This process is repeated until all nodes have heard the Floodcast.

Floodcasting is designed to facilitate large, continuous data transfers. The selection and possible subsequent re-selection (arbitration, contention) of the initiator is outside the scope of Floodcasting's design and is left to other protocols. Similarly outside of Floodcasting's design is any notion of power. Duty cycling does not make sense in the face of a continuous

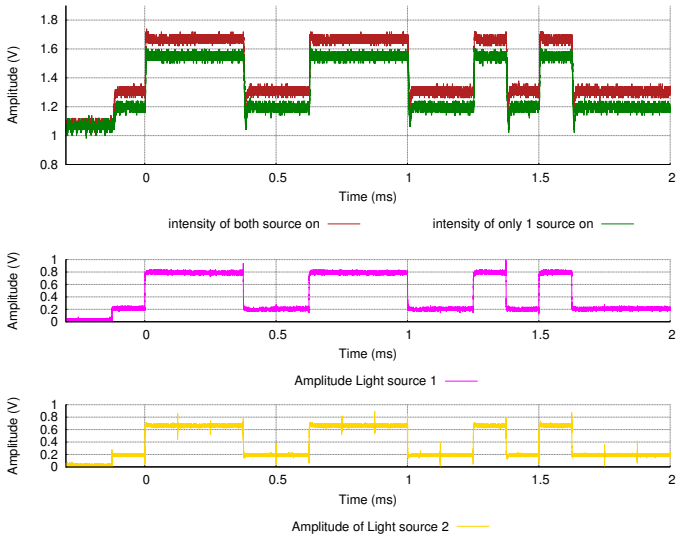


Figure 4.4: *Perceived Light Intensity and Source Synchronization*—The bottom two sub-figures are output voltages from two nodes. The output voltages control a PWM generator to modulate the duty cycle of an LED. These two nodes are synchronized by Floodcasting. The top figure shows perceived amplitude at a receiver. The output voltage indicates the light intensity in log scale. The output voltage of two synchronized light source is greater than for a single light source. In addition, this figure shows the baseline voltage of office lighting condition, which is slightly above 1 V.

stream of data that is saturating the communication channel. In designing Floodcasting, we seek to optimize the following channel features: bandwidth, latency, local buffer size, reliability, and synchronization.

Figure 4.4 demonstrates synchronized VLC transmissions using Floodcasting. We put our VLC receiver frontend in a room with normal office lighting conditions. Under these conditions, the receiver frontend outputs ˜1.05 V. In this scenario, the smart luminaires are only being used to transmit data, not to provide normal illumination. The smart luminaires are synchronized using Floodcasting in a multi-hop network. Transmitted 1s and 0s are represented by two different voltage levels which are mapped into two light intensities. Figure 4.4 shows the difference in perceived light intensity between two synchronized light sources and only one light source. Since the output voltage is linearly proportional to the log of the input intensity, the difference in output voltage is much smaller than the difference in input intensity.

4.5 Concurrent Transmission

The key idea of Floodcasting is to concurrently transmit identical packets. This only works if these packets constructively interfere with each other. This idea has been explored by many researchers recently. Dutta et al. proposed a receiver initiated protocol using concurrently transmitted hardware ACKs [27, 29]. These hardware generated ACKs are short, highly synchronized packets. These packets constructively interfere so that a receiver still has high probability to correctly decode the packet. Ferrari et al. use software techniques to synchronize nodes to concurrently transmit data packets [32, 33]. However, their software synchronization technique is worse than the hardware synchronization used in Dutta's work and Floodcasting. Furthermore, Ferrari's work could only send 8 Byte packets, limiting their bandwidth. In the following sections, we will discuss some basics of wireless communication, important factors for constructive interference, and the corresponding math models. First, we will discuss the two keys to constructive interference: *phase offset* and *carrier frequency offset*.

4.5.1 Phase Offset

To understand the effect of phase offset (time delay), we can look at a situation where 2 nodes are transmitting identical data to a receiver. We assume the modulated signal from node 1 is $cos(\omega t + \theta/2)$, and the modulated signal from node 2 is $cos(\omega t - \theta/2)$. The phase offset (time delay) between the two nodes is θ. The receiver receives a stronger signal only if the phase offset is within a certain range, which can be calculated using a power relationship.

$$\left[cos(\omega t + \frac{\theta}{2}) + cos(\omega t - \frac{\theta}{2})\right]^2 > cos^2(\omega t)$$

This equation holds when the phase offset (θ) is within $0 < \theta < 2\pi/3$. In other words, two packets constructively interference when the phase offset is low enough. In the IEEE 802.15.4 encoding scheme (O-QPSK), this translates to a maximum offset of 0.67 µs. Ferrari et al. performed a simple analysis of the effect of time delay on interfering baseband signals of equal energy. Their analysis considered only the recoverability of the first two symbols of the 15.4 packet at relatively coarse granularity. For time delays of 0.25 µs, 0.5 µs, and 0.75 µs the symbol error rate (SER) was approximately 2%, 20%, and 80% respectively [33]. However, the packet reception rate (PRR) is not solely depends on the phase offset. It also depends on the packet length. In their evaluation, the packet reception rate drops from 95% to 75% as packet length increases from 8 bytes to 128 bytes.

4.5.2 Carrier Frequency Offset

The carrier frequency (ex. 2.4 GHz) of wireless communication is generated using a low frequency (˜20 MHz) clock source (crystal/oscillator) as a reference, a voltage controlled oscillator (VCO), and a phase locked loop (PLL). However, these crystals are mechanical devices and their frequencies are vulnerable to drift resulting from aging or ambient temperature changes [78]. The reference frequency drifting results in carrier frequency drifting. In a single transmitter and receiver pair, small carrier frequency mismatches have little impact on decoding. The received amplitude is slightly attenuated, but it can be solved by tuning the variable gain amplifier (VGA) in the RF receive frontend. However, when using concurrent transmissions, carrier frequency mismatches have a stronger impact on the received baseband signal due to the *the envelope effect*. This can significantly attenuate the received signal amplitude.

To better understand the envelope effect, we assume the carrier frequencies of two transmitters are f_{c1} and f_{c2} respectively, and the receiver's carrier frequency is f_{cr}. Ignoring all channel distortion, and assuming two transmitters have the same amplitude and data $(S_I(t), S_Q(t))$, the received signal $R(t)$ can be written as:

$$\begin{aligned} R(t) &= S_I(t) \times cos(\omega_{c1}t) + S_Q(t) \times sin(\omega_{c1}t) + S_I(t) \times cos(\omega_{c2}t) + S_Q(t) \times sin(\omega_{c2}t) \\ &= S_I(t)\left[cos(\omega_{c1}t) + cos(\omega_{c2}t)\right] + S_Q(t)\left[sin(\omega_{c1}t) + sin(\omega_{c2}t)\right] \\ &= S_I(t)\left[2cos(\tfrac{\omega_{c1}+\omega_{c2}}{2}t)cos(\tfrac{\omega_{c1}-\omega_{c2}}{2}t)\right] + S_Q(t)\left[2sin(\tfrac{\omega_{c1}+\omega_{c2}}{2}t)cos(\tfrac{\omega_{c1}-\omega_{c2}}{2}t)\right] \\ &= cos(\tfrac{\omega_{c1}-\omega_{c2}}{2}t)\left[2S_I(t)cos(\tfrac{\omega_{c1}+\omega_{c2}}{2}t) + 2S_Q(t)sin(\tfrac{\omega_{c1}+\omega_{c2}}{2}t)\right] \end{aligned}$$

By down conversion, the received in-phase baseband $(r_I(t))$ can be written as:

$$\begin{aligned} r_I(t) &= cos(\tfrac{\omega_{c1}-\omega_{c2}}{2}t)\{S_I(t)\left[cos(\tfrac{\omega_{c1}+\omega_{c2}-2\omega_{cr}}{2}t)\right] S_Q(t)\left[sin(\tfrac{\omega_{c1}+\omega_{c2}-2\omega_{cr}}{2}t)\right]\} \\ &= cos(\tfrac{\Delta\omega_1-\Delta\omega_2}{2}t)\left[S_I(t)cos(\tfrac{\Delta\omega_1+\Delta\omega_2}{2}t) + S_Q(t)sin(\tfrac{\Delta\omega_1+\Delta\omega_2}{2}t)\right] \end{aligned}$$

Where $\Delta\omega_1 = (\omega_{c1} - \omega_{cr})$ and $\Delta\omega_2 = (\omega_{c2} - \omega_{cr})$. Similarly, the received quad-phase baseband signal $r_Q(t)$ can be written as:

$$r_Q(t) = cos(\tfrac{\Delta\omega_1-\Delta\omega_2}{2}t)\left[-S_I(t)sin(\tfrac{\Delta\omega_1+\Delta\omega_2}{2}t) + S_Q(t)cos(\tfrac{\Delta\omega_1+\Delta\omega_2}{2}t)\right]$$

In the ideal case (no frequency mismatch, $\Delta\omega_1 = 0$, $\Delta\omega_2 = 0$), the received in-phase signal $(r_I(t))$ is purely composed of transmitted in-phase signal $(S_I(t))$. However, when the carrier frequencies are mismatched, the received in-phase signal contains both the transmitted in-phase and quad-phase signals.

To intuitively understand the effect of frequency mismatch, we assume the receiver's carrier frequency is locked to node 2 ($\Delta\omega_2 = 0$), and the frequency mismatch of node 1 is small ($\Delta\omega_1 \sim 0$). The in-phase component will dominate the received in-phase baseband. ($S_I(t)cos(^{\Delta\omega_1 t}/_2) \gg S_Q(t)sin(^{\Delta\omega_1 t}/_2)$) The received baseband signal can be simplified as:

$$r_I(t) = S_I(t) \times cos^2(\frac{\Delta\omega_1 t}{2})$$
$$r_Q(t) = S_Q(t) \times cos^2(\frac{\Delta\omega_1 t}{2})$$

From these equations, we see that the received signal is modulated by a low-frequency envelope—*the envelope effect* ($cos^2(^{\Delta\omega_1 t}/_2)$). Figure 4.5 shows a real example of concurrently transmitted ACK packets with a carrier frequency mismatch and the resulting low-frequency envelope. The envelope introduces *local minima* to the received signal. These are important as the signal amplitude is attenuated during a local minimum, which may cause the receiver to incorrectly decode the signal. Local minima in the received signal occur whenever the following condition holds:

$$2\pi \times \tfrac{\Delta f}{2} \times t = \tfrac{\pi}{2} \times n \quad n \in N$$

Prior work measured the acknowledgment reception rate (ARR) for varying numbers of concurrently transmitting neighbors, and found that with two concurrently transmitting nodes, the worst case ARR was 97% [29].

4.5.3 Mitigating the Envelope Effect

One way to mitigate the envelope effect is *Continues Automatic Gain Control (AGC)*, which involves varying the gain setting quickly to adapt the amplitude change to provide a relatively constant amplitude over the entire packet. Another approach—*Automatic Frequency Compensation (AFC)*, dynamically adjusts the receiver's carrier frequency to match the transmitter frequency. Reducing the frequency mismatch increases the period of envelope, so that the received signal amplitude is relatively constant.

4.5.3.1 Continuous Automatic Gain Control (AGC)

The AGC algorithm detects and compensates for varying signal strengths by dynamically adapting the amplitude of the raw RF signal for further processing. The available AGC resolution and responsiveness depends on the latency of the AGC controller. Highly latency devices such as the USRP platform can only do AGC on a *per-packet* basis.

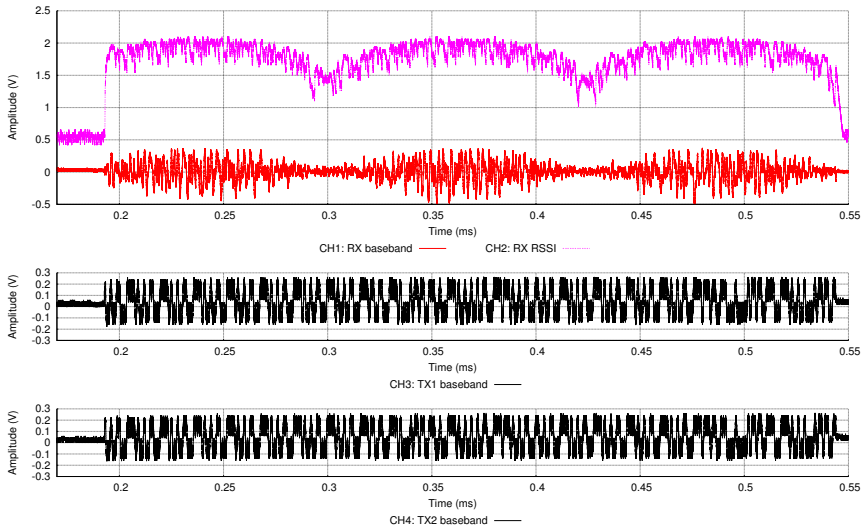


Figure 4.5: A constructive ACK collision is observed. CH1 is the RX baseband signal. CH2 is the RSSI. CH3 and CH4 are the TX baseband signals of the two colliding ACKs. The slightly offset carrier frequencies of CH3 and CH4 interact to form the envelope modulation on the received baseband signal. Without automatic gain control, envelope modulated baseband signal lowers the probability to correctly decode the packet.

Traditional AGC in commodity radios latch a gain value upon receiving the Start of Frame Delimiter (SFD). This AGC loop is sufficient if the amplitude of a signal is constant over the entire packet. However, if multiple nodes transmit concurrently, a radio with an SFD-latched AGC may experience a changing signal amplitude over time from the envelope modulation as seen in Figure 4.5. A low latency, tight control loop continuous AGC could adjust the gain setting on small time slices instead of per packet. The fast response gain control alleviates the amplitude attenuation caused by the envelope effect.

4.5.3.2 Automatic Frequency Compensation (AFC)

To better understand the frequency mismatch on a single transmitter receiver pair, we examine a transmitter *T* and receiver *R* with carrier frequencies of f_{c1}, f_{c2} respectively, and $f_{c1} \neq f_{c2}$. We denote the transmitted in-phase and quadrature-phase baseband signal are $S_I(t)$ and $S_Q(t)$. The actual signal which is being transmitted through the air, $S(t)$, can be written as follows:

$$S(t) = S_I(t) \cdot cos(2\pi \cdot f_{c1} \cdot t) + S_Q(t) \cdot sin(2\pi \cdot f_{c1} \cdot t)$$

Ignoring channel distortion, noise, and interference, the received signal $R(t) = S(t)$. The received in-phase signal can be expressed as follows:

$$\begin{aligned} r_I(t) =& R(t) \cdot cos(2\pi \cdot f_{c2} \cdot t) \\ =& [S_I(t) \cdot cos(\omega_{c1} t) + S_Q(t) \cdot sin(\omega_{c1} t)] \cdot cos(\omega_{c2} t) \\ =& S_I(t) \cdot cos(\omega_{c1} t) \cdot cos(\omega_{c2} t) + S_Q(t) \cdot sin(\omega_{c1} t) \cdot cos(\omega_{c2} t), \end{aligned}$$

where $\omega_{c1,2} = 2\pi \cdot f_{c1,2}$. Simplifying the equation by the product-to-sum identities yields:

$$\begin{aligned} r_I(t) =& \tfrac{1}{2} S_I(t) \left[cos((\omega_{c1} - \omega_{c2})t) + cos((\omega_{c1} + \omega_{c2})t) \right] + \\ & \tfrac{1}{2} S_Q(t) \left[sin((\omega_{c1} - \omega_{c2})t) + sin((\omega_{c1} + \omega_{c2})t) \right] \end{aligned}$$

By removing high frequency components ($\omega_{c1} + \omega_{c2}$), the received in-phase/quadrature-phase baseband signal $r_{IB}(t)$, $r_{QB}(t)$ can be expressed as follows:

$$\begin{aligned} r_{IB}(t) &= \tfrac{1}{2} S_I(t) \cdot cos(\Delta\omega t) - \tfrac{1}{2} S_Q(t) \cdot sin(\Delta\omega t) \\ r_{QB}(t) &= \tfrac{1}{2} S_I(t) \cdot sin(\Delta\omega t) + \tfrac{1}{2} S_Q(t) \cdot cos(\Delta\omega t), \end{aligned}$$

where $\Delta\omega = \omega_{c2} - \omega_{c1}$. By rearranging these equations, we get the following equation:

$$\begin{bmatrix} r_{IB}(t) \\ r_{QB}(t) \end{bmatrix} = \begin{bmatrix} cos(\Delta\omega t) & -sin(\Delta\omega t) \\ sin(\Delta\omega t) & cos(\Delta\omega t) \end{bmatrix} \times \begin{bmatrix} {}^1\!/_2 \cdot S_I(t) \\ {}^1\!/_2 \cdot S_Q(t) \end{bmatrix}$$

From the equation above, we see that frequency mismatches rotate the complex coordinate with angular velocity $\Delta\omega$. Hence, by measuring the rotation speed and direction at fixed intervals, we can back calculate the carrier frequency offset between the receiver and transmitter. In O-QPSK modulation, the I/Q signal is offset by ${}^\pi\!/_2$, and $I^2(t) + Q^2(t) = 1$ at any given time. More importantly, the symbol will always rotate exactly 90°on the constellation every 0.5 µs (ex. $(1,0)_{I,Q} \rightarrow (0,1)_{I,Q}$ or $(0,-1)_{I,Q}$). Measuring both clockwise and counter-clockwise rotation angles at fixed intervals (0.5 µs) allows the system to determine whether the receiver's carrier frequency is leading or lagging. With this information, we are able to adjust the receiver's carrier frequency in real time to align with the incoming packet.

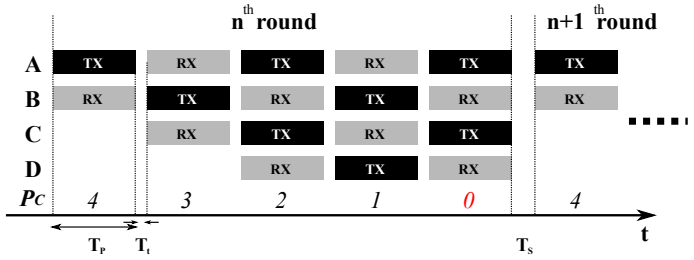


Figure 4.6: *Floodcasting*—Nodes A, B, C and D are all 1-hop from one another, connected linearly. In this example, node A initiates a flood counting down from $P_C = 4$. At first, only node B receives the packet which it re-broadcasts after a turnaround time T_t with counter value $P_C = 3$. This packet is then re-broadcast by nodes A and C with $P_C = 2$. This packet constructively interferes at node B and is simply received at node D. The re-broadcasting repeats until nodes A and C transmit a packet received by nodes B and D with $P_C = 0$. Upon completion of transmission/reception of the $P_C = 0$ packet, nodes A, B, C, and D simultaneously process the data packet.

4.6 Floodcasting Design

Floodcasting is inspired by the Glossy flooding protocol, which periodically generates flooding events [32,33]. To begin a flood, an initiator sends the first packet, which contains the data to be broadcast and a counter c. In addition, each node keeps a local counter n_{tx} that is reset to 0 at the beginning of every flood. Upon receiving a packet, nodes increment the packet counter c and immediately re-transmit the packet. Nodes also increment their internal counter n_{tx}. As a configuration parameter, all Glossy nodes have a maximum number of transmissions N per flood event. Once a node's n_{tx} reaches N, the node returns to sleep. Once all nodes have transmitted N times, the Glossy flood is complete. However, Glossy is limited to sending 8 bytes packets because of low reception rates for longer packet due to the envelope effect. In this section, we discuss the design of Floodcasting, which utilizes AGC to mitigate the envelope effect.

Compared to traditional multi-hop networks, Floodcasting trades a reduction in bandwidth and possible increase in latency for higher reliability and tighter synchronization. The flood release event does not depend on the local clock of any node. Floodcasting can be imagined as a cheer in a stadium that alternates even and odd sections yelling "Go" and "Team". It begins when section 10 decides to initiate by yelling "Go". Sections 9 and 11 respond back "Team". Section 10 hears sections 9 and 11 yelling together while sections 8 and 12 hear 9 and 11 respectively. Soon the entire stadium is alternating, with very tight precision, "Go" and "Team". To achieve the countdown effect, the initiating section would instead begin by yelling a number, and each echo back yells one number lower. Since the sections agree on the rules for modifying the cheer, they will constructively interfere as

they count down. With a high enough starting number, soon the entire stadium is yelling, with alternating sections yelling even and odd numbers. Eventually, half of the stadium is yelling "0" and the other half is listening to them yell "0". After either yelling or hearing "0", the entire stadium can yell "Go Team!" together in a highly synchronized manner.

Translating the analogy to the actual Floodcasting design, Figure 4.6 shows a Floodcasting started by node A with a gate counter $P_C = 4$. With Floodcasting, the counter value P_C provides a tunable knob between latency and reliability. To ensure the entire network receives a flooding packet, P_C must be set to at least the maximum number of hops h from the initiator. Increasing P_C beyond h will increase the number of times that the network echoes the packets internally, which helps to increase the probability that an unreliable link receives at least one of the flood's packets. As a further element of robustness, nodes in a Floodcasting will also count down using their local clock from the last flood packet they received, so that if they experience a transient disconnection near the end of the countdown, they will still participate in the flood release, albeit with slightly greater jitter. One architectural difference between Floodcasting and Glossy is Floodcasting guarantees the network is quiet once the counter reaches 0, whereas Glossy depends on the diameter of the network. This provides a deterministic timing for a packet to live in the network so that nodes can initiate a new packet accordingly.

4.7 Floodcasting Analysis

Bandwidth. The maximum bandwidth available to Floodcasting scales with the network size. With Floodcasting, at any given time, all the packets in the network are identical. Eventually, the entire network must possess the same packet, therefore the larger the network the lower the bandwidth. We can express the bandwidth R available to a Floodcasting channel with h hops as:

$$R = \frac{\text{Payload Bytes}}{h \times \text{Packet Bytes} \times T_{byte} + T_t},$$

where T_t[1] is radio turn over time; T_{byte} is the time duration for a byte (32 µs); maximum packet length and maximum available payload size are 127 Bytes and 116 Bytes respectively. Substituting these values in, we can find an expression for the maximum bandwidth R_{max} as a function of the network hop count h:

$$R_{max}(h) = \frac{116(\text{Payload Bytes})}{h[127 \times 32\ \mu\text{s} + 192\ \mu\text{s}]} = \frac{26.08}{h}(\text{kBytes/s})$$

[1]Minimum 192 µs in the 802.15.4 standard.

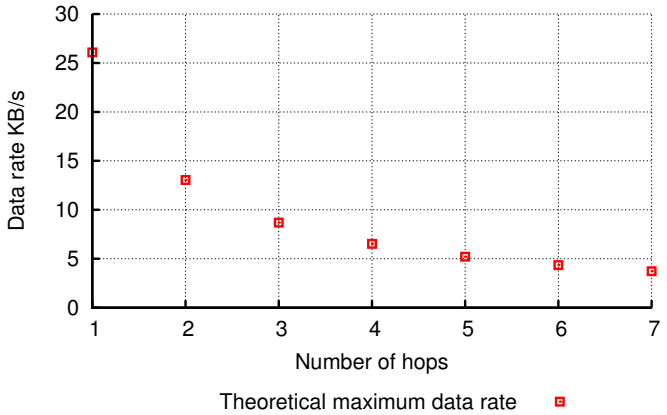


Figure 4.7: Maximum available bandwidth in a h hops Floodcasting network. This figure shows that the maximum available data rate drops when the diameter of the network increases.

Which we can re-write as:

$$R_{max}(h) \times h = 26.08\text{(kBytes/s)}$$

This theoretical upper bound for bandwidth is limited by the diameter of the network. Figure 4.7 plots available bandwidth versus hops. This relationship is modeled as a hyperbolic function. The initial bandwidth cost of adding additional hops to the network is large, but the relative cost of additional hops to a moderately sized network is comparably small.

Although the available bandwidth is low for large network diameters, Floodcasting can provide tight *time synchronization* to networked smart luminaires. We envision the bulk of data transfers to smart luminaries to occur on the data plane, with Floodcasting being dedicated to support synchronized VLC transmissions.

Latency. The latency of Floodcasting is measured from when the initiator sends out a packet to when the gate counter (P_c) reaches 0. Latency is a function of the counter value, packet length (T_p) and radio turn over time. However, the flood gate counter (P_C[2]) embedded in the initial packet defines the boundary of each flood round. This also determines the latency (L):

$$L(N) = T_p \times P_C + T_t \times (P_C - 1)$$

Buffer Size. In Floodcasting, all the packets in flight during a flood are identical. Therefore, each node in the network only requires 1 packet-sized buffer to operate.

[2]For proper operation, $P_C \geq h$

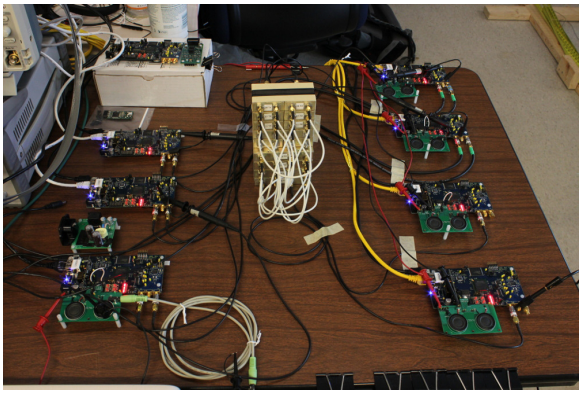


Figure 4.8: Flexible smart luminaire testbed array. The box in the middle of the picture is a stack of power dividers. The bottom left smart luminaire is the initiator. This example uses Floodcasting to provide synchronous realtime audio across the multihop platforms.

4.8 Implementation

To test and evaluate Floodcasting, we implemented Floodcasting on the FPGA on our experimental smart luminaire platform [51]. Figure 4.8 is a picture of a 5 node deployment of our testbed, with one initiator node and four other Floodcast nodes. Our Floodcasting implementation, coupled with the rest of the 802.15.4 stack, filled the available FPGA area. We implemented the continues automatic gain control as we described in §4.5.3.1, but did not have sufficient space to include automatic frequency compensation.

4.8.1 Smart Luminaire Testbed

Previous work exploring constructive interference went to great pains to coordinate the latency of the radio control loop [27, 33]. By leveraging the flexible RF frontend on our testbed, we are able to build our Floodcasting deployment with greater precision than any commercial radio and microcontroller combination could allow. An IEEE 802.15.4 GNU Radio implementation is also available for the USRP [81]. However, this implementation does not support a low-latency control path as is available on our testbed. For example, the proposed continues AGC cannot be ported to the USRP. A USRP implementation would also have required a PC to operate, which has an unpredictable interrupt latency, making it more challenging to meet Floodcasting's precise timing constraints.

4.8.2 Floodcasting Implementation

In our implementation, a packet with a specific destination address (0xfffe or 0xfffffffffffffffe) is considered a Floodcasting packet. The data sequence number (DSN) field of a flood

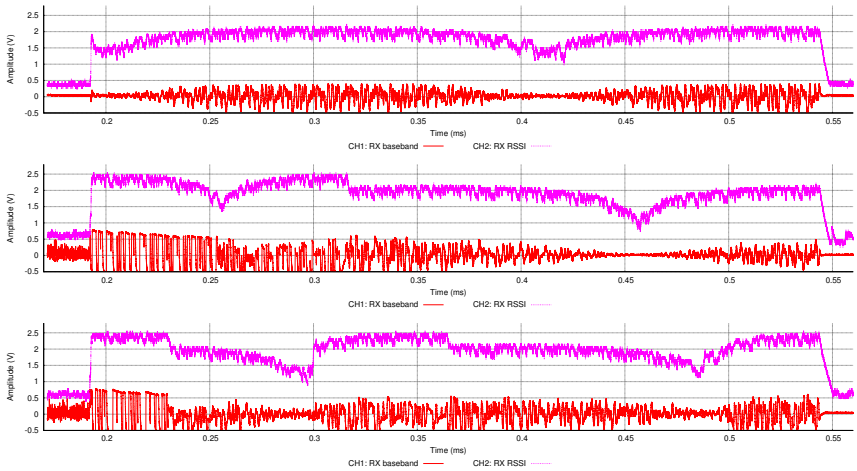


Figure 4.9: The envelope effect with different AGC methods. The top figure shows the receive baseband with AGC disabled, and amplitude is attenuated at local minimums. The middle figure shows a SFD-latched AGC, which is commonly used in a commercial radio. The radio locks the gain when it detects an SFD, but amplitude still attenuates. The bottom figure shows a continues AGC, which provides a fixed latency gain control that constantly adjusts the gain during a packet reception. Continues AGC can mitigate the attenuation caused by the envelope effect.

packet acts as the flood gate counter P_C.

When a packet is received, P_C is decremented, and the packet is retransmitted all without interrupting the microprocessor. This allows our implementation to maintain a very precise packet turnaround time T_t of exactly 192 µs.

The microprocessor is interrupted to indicate a flood gate event whenever a node either transmits or receives a packet with $P_C = 0$. Additionally, every time a node receives a packet it sets a timer to $P_C \times (T_P + T_t)$. If this timer expires, the node will assume the final packet $P_C = 0$ was dropped and trigger a flood gate event.

4.9 Evaluation

4.9.1 Continuous Automatic Gain Control (AGC)

We evaluate AGC by measuring its effect on acknowledge reception rate (ARR). We transmit 50,000 packets to two nodes with the ACK field. One node uses an SFD-latch AGC, while the other uses continuous AGC. Figure 4.9 shows the received baseband signal (Red) and RSSI (Purple) under i) no AGC, ii) SFD-latched AGC, and iii) continues AGC. Table 4.1 compares the two AGC methods. The results show that continuous AGC offers a small improvement in ARR over SFD-latched AGC.

AGC MODE	AVERAGE ARR	STD
SFD-latch	97.27%	0.211%
Continuous	98.20%	0.065%

Table 4.1: Acknowledgment reception rate (ARR) for two constructively interfering transmitters with respect to different AGC modes. We transmitted 10,000 ACKs per transmitter per experiment and repeated each experiment 5 times. A continuous AGC performs slightly better due to a smaller carrier frequency separation between the transmitting nodes.

Further experimentation with AGC reveals that the properties of each AGC method rely heavily on the degree of separation between the two carrier waves. Figure 4.10 plots the reception rate of concurrently transmitted packets against varying differences in the two transmitting carrier wave frequencies. Recall that the period of the beat frequency of the modulating envelope is given by $T = \frac{1}{\Delta f}$. For lower values of Δf, this means the period of envelope beats will be relatively long and the continuous AGC is able to adapt and correct for the variation in signal strength. As Δf increases, however, the local minima from the envelope wave increase in frequency (while decreasing in length). Eventually the continuous AGC cannot adapt fast enough to the varying signal strength imposed by the envelop.

Once the AGC loop latency is greater than the beat period, the minima are so short that only a few chips are lost. The 802.15.4 protocol employs a spreading technique such that 4-bits of data are composed into 1 symbol made up of 32 chips. The redundancy supplied by the spreading means that once the beat frequency of the envelope is too high to correct via AGC, only a few chips of the symbol are dropped, allowing the symbol as a whole to be correctly decoded. This phenomena explains the observed upward trend of the SFD-Latch AGC as Δf increases.

As Δf grows sufficiently large, the frequency of the minima begins to approach and ultimately surpass the speed of the continuous AGC control loop. The loop delay of the continuous AGC calculation then causes the actual gain to be applied to the incoming signal too late. The extra strength oscillation imposed by the late gain adaptation accounts for the $1{\sim}2\%$ worse performance of continuous AGC versus the SFD-latched AGC for high Δfs.

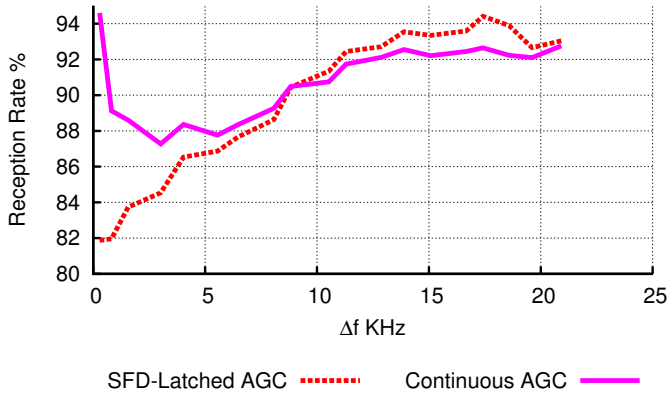
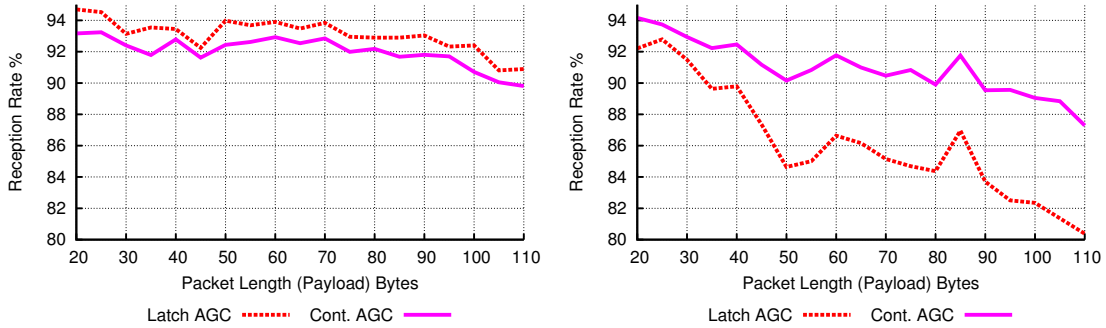


Figure 4.10: Reception rate versus carrier frequency separation of two concurrent transmitters with a fixed packet length (60 bytes). The period of the beat frequency of the enveloping modulation is $T = \frac{1}{\Delta f}$. For small Δf, this period is sufficiently long that continuous AGC is able to correct for the varying signal strength (compare Figure 4.5 CH3). As Δf grows, the beat period shortens until it is too fast for continuous AGC to keep up. At this point, however, the minima are sufficiently narrow to only obscure a few chips and the spreading built into 802.15.4 recovers the missing information. The continuous AGC's attempt to follow the high-frequency minima account for for the slightly worse performance of continuous AGC at higher Δfs. This illustrates how low-level control can improve protocol performance.



(a) RF frequencies separate by 16.5 kHz

(b) RF frequencies separate <1 kHz

Figure 4.11: Reception rate of constructively interfering packet collisions with the transmitter at two slightly different carrier frequencies. Both transmitters send the exact same message, at the same time. Figure 4.11a shows the result when the transmitters are separated by 16.5 kHz, while Figure 4.11b depicts the case of the transmitters separated by <1 kHz. In both cases, the longer the packets, the lower the reception rate as we get more beats in a single packet. This leads to a lower signal amplitude, and thus potential for decoding errors. To mitigate this, the AGC can be held constant after latching it at the SFD detection, or it can continuously updated over the whole packet length. For small carrier frequency offsets (Figure 4.11b) continuously updating the AGC improves the reception rate by 2˜6%.

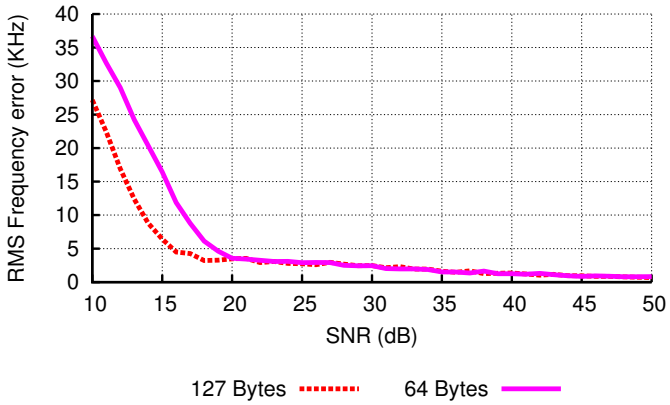


Figure 4.12: AFC simulation. Frequency separation of TX and RX is 50 kHz initially ($f_{c1} - f_{c2} = 50$ kHz). The figure shows the RMS frequency separation after receiving a packet with AFC enabled. As expected, AFC performance improves with higher SNR and greater packet length.

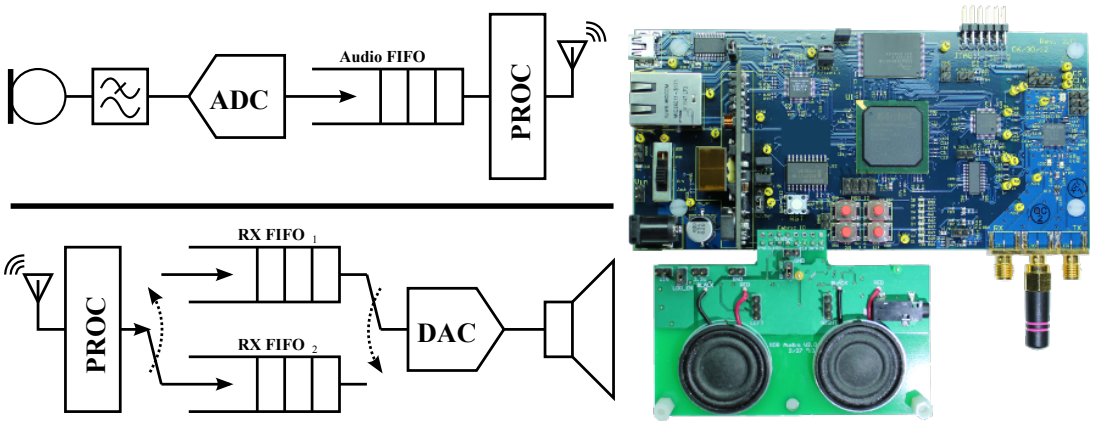
4.9.2 Automatic Frequency Compensation (AFC)

Figure 4.12 shows the simulated result of running AFC on our experimental platform. In this simulation, f_{c1} and f_{c2} are set 50 kHz apart and the step size of f_{c2} is 300 Hz. After receiving a packet, the AFC adapts the receiver carrier frequency f_{c2} to reduce the frequency mismatch. Two factors are important for AFC effectiveness: packets with higher SNR afford more accurate angle estimation and packets with longer payloads provide more opportunities to adapt the carrier frequency.

In Table 4.2, we explore the combination of AFC with the previously explored AGC. Combining the continuous AGC with AFC yields the best result, a 95.5% ARR. Applying AFC to the simpler SFD-latched AGC results in a 1.2% reduction in ARR, however. Recall

AGC MODE	AFC MODE	ARR
SFD-latch	Enable	93.3%
Continuous	Enable	95.5%
SFD-latch	Disable	94.5%
Continuous	Disable	95.1%

Table 4.2: Acknowledgment reception rate (ARR) for two constructively interfering transmitters with respect to AFC and different AGC modes. We transmitted 10,000 ACKs per transmitter per experiment. With AFC disabled, carrier frequencies of TX and RX were off by 16.4 kHz. In both cases, continuous AGC worked better. Enabling AFC worsened the reception rate for SFD-Latch AGC because the baseband signal is attenuated by the low-frequency envelope.



(a) Architecture of audio sampling/playback.

(b) Audio streaming on SDL.

Figure 4.13: The upper half of Figure 4.13a shows the sampling subsystem that is composed of audio frontend and a buffer. Bottom half is receiver architecture. The double buffers provide seamless playback. Figure 4.13b shows our experimental setup. The bottom add-on is our audio sampling and play-back system. The audio board has a 16-bit ADC, DAC and audio amplifiers.

from the AGC discussion and Figure 4.10 that SFD-latched gain control performs poorly for small Δf as the local minima from the envelope beats are too long. As the AFC attempts to compensate for the frequency difference between TX and RX, these minima become longer which improves the performance of the adaptive AGC, but results in more below-threshold signal windows for the fixed-gain, SFD-latched method.

4.9.3 Flooding Music Real-time

In this section, we demonstrate the high flooding bandwidth of Floodcasting by flooding real-time audio samples in a multi-hop network.

Figure 4.13a shows the architecture of our real-time audio application. The upper half is an audio sampling sub-system. The audio FIFO temporarily stores the audio samples. Once the previous flooding round is completed, the node uploads the samples the from audio FIFO to the TX FIFO until the TX FIFO has enough data bytes. Then the node initiates a new round of flooding. The play-back subsystem, however, has two RX FIFOs and employs a double buffer to mitigate the different data rates of wireless packets and audio samples.

We've shown the maximum available bandwidth of Floodcasting is $^{26.08}/_{h}$ kBytes/s. In our experimental setup, we set $h = 3$, $F_s = 8$ ksamples/s, and 1 byte/sample. Given these parameters, the minimum length of a packet is 77 Bytes. The payload size can be increased

to allow some slack between each flooding round and more efficient channel usage because of fixed non-data overhead. However, longer packets require larger TX/RX FIFOs which increases the hardware area. More importantly, Floodcasting is based on multiple packet collisions. Even with continuous AGC, longer packets lower our reception rate.

4.10 Discussion

For Floodcasting to work properly, the initiator must estimate the size of network h to assign an appropriate counter value. This value affects *bandwidth, latency,* and *reliability*. *Bandwidth* is inversely proportional to the network size h, which implies that Floodcasting cannot provide sufficient bandwidth for large networks. Furthermore, the number of transmissions is not equal throughout the network. The number of transmission for a node is a function of its the distance from the initiator. Closer nodes transmit more frequently than further nodes, which means that power usage is a gradient. The *latency* in Floodcasting, however, is static. Latency has a strict relationship with *bandwidth* and network size h. This reliability comes at an efficiency cost. Setting the flood gate counter too close to the network size leads to the edge nodes only having a limited number of chances to receive the packet. The *reliability* of edge nodes suffer in the face of efficiency.

To the best of our knowledge, no commercial radio is equipped with packet forwarding capability. Thus, utilizing Floodcasting requires a software defined radio. Also, commercial radios do not allow designers to adaptively tune the gain or carrier frequency in a tight control loop or with fine granularity, which reduces the reception rate for constructive interference based systems such as Floodcasting.

4.11 Time Synchronization as an SDL Service

Energy is one of the most important constraints in sensors nodes. Periodic battery replacement/recharging requires labor, time and money. Hence, many sensor nodes are designed with low-duty cycles and/or use energy harvesting to preserve energy or eliminates periodic battery charging. In a wireless network, time stability determines the length of the guard band radios use for neighbor discovery and data exchange. Since wireless radios are the major energy consumers in sensor nodes, shortening the radio guard band can significantly extend battery life.

Today, off-the-shelf oscillators have ˜5 uW power draw and $>$100 ppm stability over their operational temperature range, which is not optimal for ultra-low power systems. On the other hand, researchers have demonstrated a sub nW oscillator with custom silicon

that trades stability and temperature sensitivity (31 ppm/°C) for power [55]. Although the oscillator itself draws little power, the increase needed in radio guard time offsets this gain.

Before high stability, nano-power oscillators become available, using synchronized event triggers may help nodes in a network stay synchronized, keeping radio guard times short.

Because Floodcasting allows networked smart luminaires to send tightly synchronized transmissions, SDL can offer time synchronization as a service to sensor nodes by providing synchronized VLC wakeup sequences. Thus, instead of using high quality, high power timers, nodes can use an ultra-low power optical wakeup receiver to stay synchronized [43]. Once the VLC wakeup sequence is received, the optical frontend asserts an interrupt to wake up the processor. Nodes can then perform wireless neighbor discovery or data exchange with short radio guard bands.

4.12 Summary

In this chapter, we discussed the architecture of diffusing VLC receivers, the challenges in communicating with them, and how we solved those challenges. We introduce **Floodcasting**, a data dissemination protocol supporting real-time actuation and control. Floodcasting is a flooding protocol that provides sub-µs synchronization, and up to 26 kBytes/s bandwidth across a multi-hop network. The design of Floodcasting is based on constructive interference of wireless packets. We derive and verify the mathematical model for frequency mismatches between transmitters and receivers, and propose several techniques to mitigate the effect caused by frequency mismatches. We also demonstrate synchronized VLC transmissions from smart luminaires using the time synchronization provided by Floodcasting. In addition to increasing transmission range, synchronized smart luminaires are able to communicate with diffusing VLC receivers using simple on-off keying data modulation. Unsynchronized smart luminaires would need to implement a multiple access scheme to communicate with diffusing VLC receivers. Furthermore, synchronized smart luminaires are able to provide provide time synchronization as a service to sensor nodes with VLC receiver frontends. We now explore how to communicate data to a camera receiver—an array device that relaxes the synchronized transmission requirements of a diffused received but requires more sophisticated image processing to decode transmitted VLC signals.

CHAPTER 5

VLC to a Camera Receiver (CamComm)

In this chapter, we explore VLC communication to camera receivers. Unlike diffusing receivers in §4, a camera receiver is an array of photodiodes with a color filter. Charge-coupled device (CCD) and complementary metal-oxide-semiconductor (CMOS) are the two commonly used imager types, with CMOS imagers dominating the low-end imager market. One unique property of CMOS imagers is the rolling shutter effect, caused by capturing the frame one row at a time. The rolling shutter effect results in motion blur, but it enables imagers to receive data from blinking lights.

5.1 Key Properties

In this section, we discuss the unique issues of using a camera as a VLC receiver rather than as a photodiode. Also, we focus the use of low-cost, ubiquitous CMOS image sensors as opposed to CCD image sensors.

Rolling Shutter. When capturing an image, a CMOS imager exposes one (or more) columns of pixels at a time, but the pixel structure allows reading out only one column at

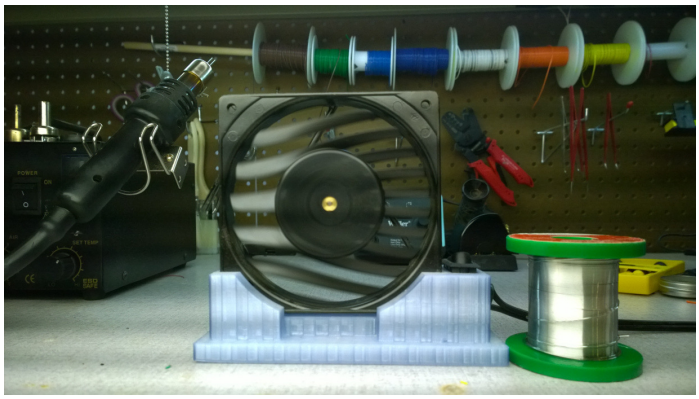
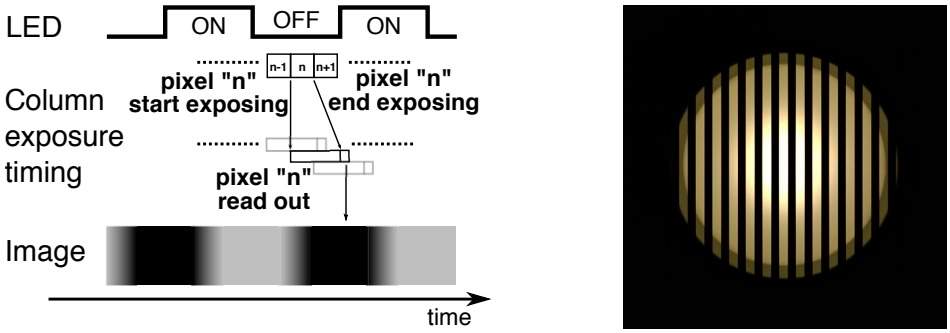


Figure 5.1: Motion blur due to the rolling shutter effect on a CMOS camera. This image is captured with 1/1000 s exposure time. The high speed rotation of the fan's blades is distorted.



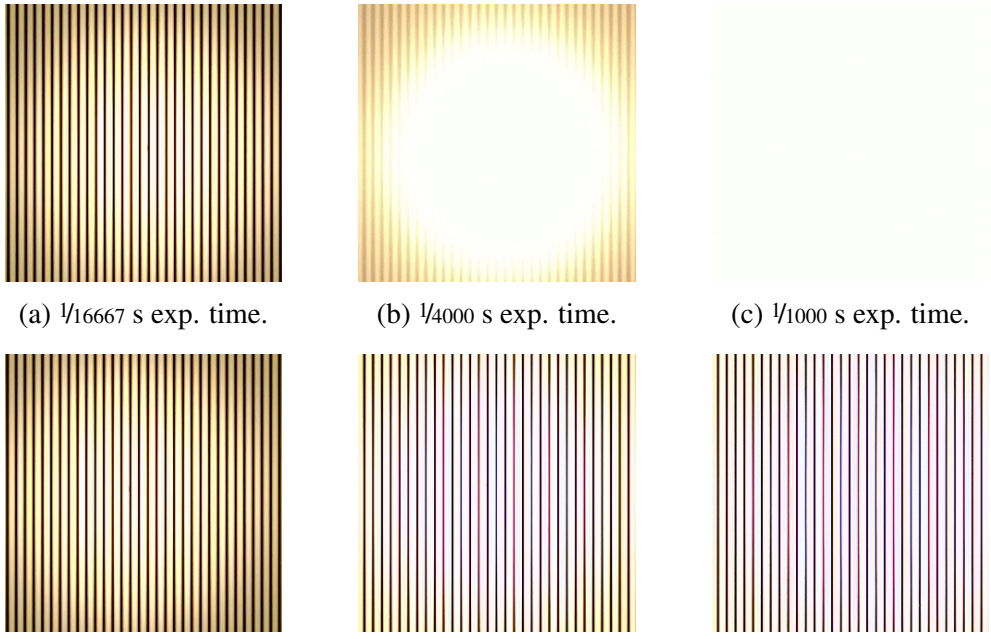
(a) Rolling shutter principle.

(b) 1 kHz pure tone.

Figure 5.2: Figure 5.2a illustrates how CMOS cameras capture a frame column by column. Adjacent columns are exposed to light with a time offset. Thus, the imager captures a distinctive banding pattern from a rapidly flashing LED. Adjusting the LED's frequency or duty cycle results in changes to the width of the light and dark bands in the image, allowing frequency to be detected and decoded. Figure 5.2b shows the captured image when using a mobile phone camera to take a picture of an LED flickering at 1 kHz with a 50% duty cycle. with 50% duty cycle.

a time. To capture a picture, the hardware sweeps across the image one column at a time, effectively creating a "rolling shutter." This architectural "defect" results in motion blur while taking a picture of a high speed moving object. Figure 5.1 shows the image captured using a CMOS camera with $1/1000$ s exposure. Motion blur due to the rolling shutter effect can be observed in fan's blades. Similarly, when imaging a rapidly oscillating light source, the rolling shutter results in a distinctive banding effect, as Figure 5.2 shows, in which some columns capture the light in the on state, some in the off state, and some in a transition state. The column readout frequency is called the scan rate, and is a resolution-dependent property of the imager.

Exposure Control. Exposure control determines how long a pixel collects photons. During exposure, a pixel's charge accumulates as light strikes, until the pixel saturates. The relationship between exposure time and LED flicker period determines the amplitude of the banding pattern. Imagine the case when the exposure time is much greater than the flicker period. Every pixel collects a similar amount of photons, leading to a small amplitude of the bands. Also, if the intensity of the transmitter is high, a long exposure time might saturate all of the pixels. Another scenario is an LED flicker period greater than or equal to the exposure time. Under this condition, every pixel collects exactly the same amount of photons so that there is no banding effect. The useful case is when exposure time is much shorter than the flicker period. Under this scenario, some pixels could be completely dark, which enlarges the signal amplitude. Figure 5.3 shows images captured



(a) 1/16667 s exp. time. (b) 1/4000 s exp. time. (c) 1/1000 s exp. time.

(d) ISO 100. (e) ISO 400. (f) ISO 800.

Figure 5.3: Effect of exposure time and film speed. The transmitter is set to flicker at 1 kHz. Shorter exposure images show a clear banding pattern whereas longer exposure images do not. Figure 5.3b also suggests that longer exposure rates saturates the imager. Film speed controls gain of the amplifier, so higher film speeds show higher average brightness.

under different exposure settings. A banding pattern can be observed with short exposure times. However, longer exposure times have a lower contrast and the potential to saturate the image.

Film Speed. Film speed (ISO) determines the sensitivity of the image sensor by controlling the gain of the amplifiers, which boost the signal before it is sampled by the ADC. Loosely, it is a measure of how many photons are required to saturate a pixel. A faster film speed (higher ISO) saturates the image sensor with fewer photons. Thus, faster film speed is used in low illumination environments. Similarly, faster film speed could enhance the image contrast and potentially enlarge the decoding area when the receive signal is weak. We prefer smaller ISO values due to the proximity and brightness of indoor lights. Figures 5.3d to 5.3f demonstrates faster film speed is brighter under the same exposure setting.

Exposure Control and *Film Speed* are two fundamental parameters that determine the decodability of the image. To quantify image contrast using the rolling shutter effect under different exposure and ISO settings, we use a Commercial Electric T66 LED luminaire and a Nokia Lumia 1020 placed 1 m apart, with the LED luminaire configured to toggle

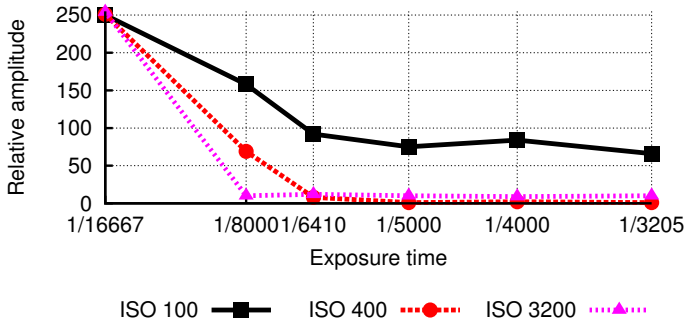


Figure 5.4: The ratio between the brightest and darkest pixels in an image. The longer the exposure, the higher the probability that a pixel saturates, reducing the resulting contrast between the light and dark bands. Hence, we minimize the exposure time to maximize the contrast ratio. Faster film speed corresponds to higher gain setting of amplifiers. As a result, the number of photons required to saturate a pixel is fewer under higher film speed.

at 1 kHz. We sweep the available exposure and ISO settings on the phone and convert the image into 8-bit gray scale to measure the banding amplitude. Figure 5.4 indicates all three ISO settings have a large signal amplitude at minimum exposure ($1/16667$). As the exposure time increases, a higher ISO saturates the image quickly whereas a low ISO still maintains certain amplitude.

5.2 Encoding Schemes

Camera communication refers to using camera as a receiver. In §2, we visited screen-to-camera communication. In this section, we will only discuss the modulation scheme used for LED luminaire to CMOS camera communication which leverages the rolling shutter effect. The common techniques in this domain are multiple frequency shift keying (MFSK) and on-off keying (OOK) with Manchester encoding.

MFSK uses multiple frequencies to encode symbols and can be decoded using simple image processing such as peak finding or FFT on a sliding window. The decoding process is light and can be performed directly on mobile devices. The available data rate for MFSK is determined by the number of observable symbols and the number of bits per symbol, which is related to the frequency decodability.

OOK with Manchester encoding uses the transition of a pulse to encode information, allowing for constant luminaire brightness. To decode data, the receiver has to first recover the clock before decoding any symbols, which is more resource consuming than an FFT for mobile devices. Alternatively, we can recover the clock using an FFT of an idle pattern interspersed between packets. The idle pattern between each Manchester encoded packet

(a) Frequency Encoding. 1 kHz tone, 50% duty cycle. (b) Manchester Encoding. Repeating data (0x66). (c) Hybrid Encoding. 3 kHz data with 6 kHz tone.

Figure 5.5: The effect of the CMOS rolling shutter and various encoding schemes. All images are taken by a Lumia 1020 of a modified Commercial Electric T66 6 inch (10 cm) Ceiling LED. The camera is 1 m from the LED and pictures are taken with the back camera. The images shown here are a 600×600 pixel crop focusing on the transmitter. The ambient lighting conditions remained the same across all images.

needs to occupy a spectrum that is distinct from the spectrum of the packets, for example a 1 kHz Manchester code with a 2 kHz pure tone as the idle pattern.

Figure 5.5 shows different encoding schemes. Figure 5.5a encodes data using pure tone frequency and Figure 5.5c shows a hybrid encoding that uses an idle pattern between packets at twice the frequency of the encoded data.

5.3 Smartphone Camera

Image resolution, **Rolling shutter scan rate**, **Platform support** and **Focal length** are key parameters for a camera. Image resolution provides more detail at a fixed viewing angle, and scan rate sets the upper bound of the bandwidth. The rolling shutter scan rate can be roughly estimated by multiplying FPS and width (or length) of the image sensor.

We emphasize that the platform support is not a hardware issue but a software issue. Exposure and ISO settings are controlled by OS-managed feedback loops. We are able to coerce these feedback loops by shining a bright light into imagers and removing it at the last moment before capturing an image of our transmitters. Using this technique, we are able to capture images with $1/7519$ s exposure on ISO 68 film using Google Glass and $1/55556$ s exposure and ISO 50 on an iPhone 5 (iOS 7). Fortunately, these low level camera configurations are gradually exported to application developer.

Focal length is the distance from the center of the lens to the focus point. This parameter is critical for indoor localization as we describe in §6. In photography, focal length is a broadly used term for distinguish lenses, and it also infers the **Field of View (FoV)** of

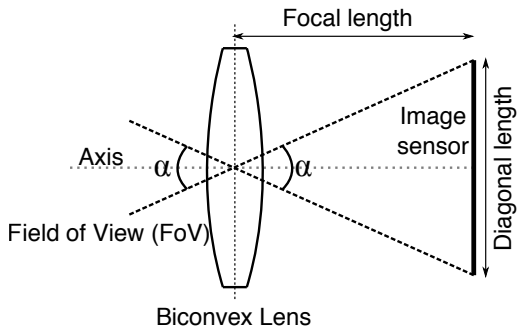


Figure 5.6: Field of View (FoV) of a biconvex lens.

lenses. Figure 5.6 shows the relationships between focal length and FoV. In general, the FoV is calculated using diagonal length of image sensor.

$$FoV = 2 \times tan^{-1}(\frac{\text{diagonal length of image sensor}}{2 \times \text{focal length}})$$

From the equation above, the FoV is a function of the size of the image sensor. Hence, lenses marked with same focal length have different FoV on different cameras. In photography, **35 mm equivalent** focal length is a metric to standardize the FoV. The diagonal length of 35 mm film is 43.3 mm (24 mm by 36 mm). The field of view can be calculated using an equivalent focal length with 43.3 mm diagonal length. For example, a 50 mm focal length (35 mm equivalent) lens would have a 46.8° FoV. For many low-end DSLR cameras, the image sensor is smaller than 35 mm film. The equivalent focal length is calculated by multiplying a **crop factor**. Since many digital cameras only report the 35 mm equivalent focal length, the real focal length can be calculated using equivalent focal length and size of image sensor. However, in a localization application, the distance in pixels, Z_f, between the center of the lens and the image sensor is a more useful parameter.

We explore camera communication using the Nokia Lumia 1020 phone mainly because it natively supports low level configurations (e.g. exposure and film speed) and has the highest resolution among all other popular phones allowing us to identify more possibilities. We use the Nokia Pro Camera application included with the Lumia, which allows the user to specify exposure and ISO settings, to capture images for this purpose. We measure the distance between the lens and imager, Z_f (1039 pixels, 5620 pixels), and scan rate (30,880 columns/s, 47,540 columns/s), for the front and back cameras respectively. To estimate the impact of manufacturing tolerances, we measure these parameters across several Lumia 1020s and find only a 0.15% deviation, suggesting that per-unit calibration is not required.

Phones	Imager resolution	Pixel size	35 mm equivalent	FoV (diagonal)	Z_f (pixels)
iPhone 5	2448x3264	1.4 µm	31.3 mm	69°	2950
iPhone 5S	2448x3264	1.5 µm	29.1 mm	73°	2750
Galaxy S4	3096x4128	1.12 µm	31.7 mm	57°	3781
Lumia 1020	4352x7712	1.12 µm	31.4 mm	69°	5620
Moto X	2432x4320	1.4 µm	28.0 mm	75°	3207
HTC ONE	1520x2688	2.0 µm	26.8 mm	77°	1910

Table 5.1: Parameters for mobile phones. The image sensor resolution of popular phones range from 8 to 33 M pixels. The HTC One has only 4 M pixels, but uses a significantly larger pixel size to enhance the image quality. The Lumia 1020 has the best image resolution among all popular phones.

The parameter Z_f is not a fixed value on a camera. A motor is used to adjust the distance between imager and lens for sharper images. This raises the question of how Z_f deviates while focusing on different distance. In a simple biconvex lens model, the relationship between s_1 (distance from object to lens), s_2 (distance from lens to imager), and f (focal length) is:

$$\frac{1}{s_1} + \frac{1}{s_2} = \frac{1}{f}$$

where s_2 and Z_f are the same parameter but s_2 is measured in meters whereas Z_f is measured in pixels. s_2 can be rewritten as $\frac{s_1 \times f}{s_1 - f}$. For the Lumia 1020, $f = 7.2$ mm. In the general use case, s_1 is on the order of meters which leads to s_2 values between 7.25 mm ($s_1 = 1$ m) and 7.2 mm ($s_1 = \infty$). This suggests that Z_f should deviate only 0.7% from a 1 m focus to infinity. As lighting fixtures are most likely 2 m to 5 m above the ground, the practical deviation is even smaller, thus we elect to use a fixed Z_f value for localization. We measure Z_f while the camera focuses at 2.45 m across 3 Lumia phones. All Z_f values fall within 0.15% of the average: 5,620 pixels.

While the front camera is more likely to face lights in day-to-day use, we use the back camera for our experiments since it offers higher resolution. Both cameras support the same exposure and ISO ranges, but have different resolutions and scan rates. Scan rate places an upper bound on transmit frequency, but the limited exposure range places a more restrictive bound, making this difference moot. Resolution imposes an actual limit by causing quantization effects to occur at lower frequencies. Given Hendy's Law—the annual doubling of pixels per dollar—we focus our evaluation on the higher-resolution imager, without loss of generality. The parameters for various popular smartphones available today are shown in Table 5.1.

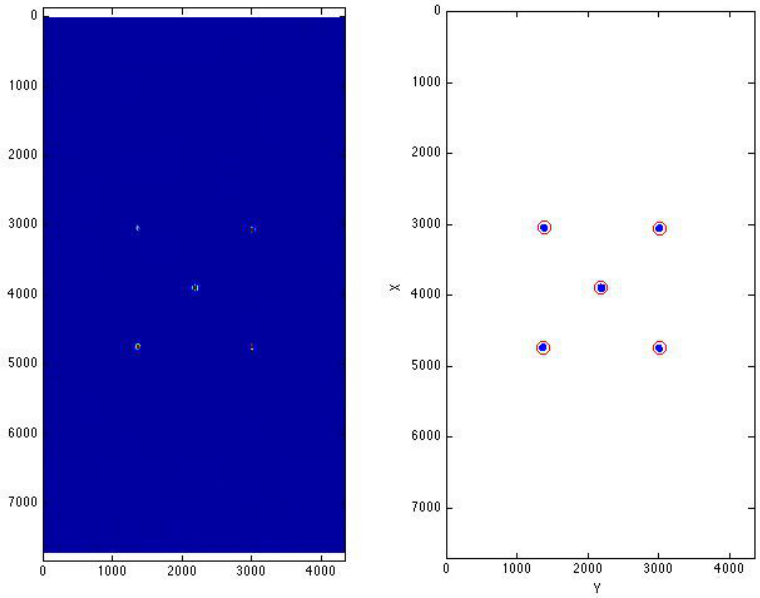


Figure 5.7: Estimated transmitters locations using K-means clustering [61]. On the left is the original image converted into gray scale. Blue dots on the right image are pixels that are brighter than 99% among all pixels. The red circles are the center of clusters of blue dots using K-means clustering.

5.4 Decoding

Independent of any modulated data, the first step is to find the centroid and size of each transmitter in the captured image. This section describes the two steps of the decoding process: finding transmitters and decoding the corresponding information. To decode the data, we explore frequency recovery and Manchester decoding.

5.4.1 Locating Transmitters

Clustering – method one. Our initial approach uses clustering technique to "count" the number of transmitters and estimate their centroids. First, the entire image is Gaussian blurred. The blurring reduces the contrast of the stripes, which resulted from the rolling shutter effect. Second step calculates the 99th percentile brightness of the entire image and labels pixels that are greater than 99th percentile brightness. The processed image now contains many groups of dots and we apply K-means clustering on those dots. To "count" the number of transmitters, we iterate the number of clusters and calculate the average distance between dots and the center of the cluster. The average distance decreases while

the number of clusters increases, but the transition is smooth once the number of clusters is greater than transmitters. We sweep the cluster size and use changes in distance to estimate the number of transmitters. The centroid of clusters is the centroid of transmitters. The major drawback of this approach is that it is computationally expensive. Sweeping the cluster size using Matlab on a MacBook Pro with an Intel i7 processor requires several seconds to complete. Figure 5.7 shows the processing pipeline of the clustering approach.

Image processing – method two. Method two uses various image processing techniques, as shown in Figures 5.8a to 5.8e, for identifying disjoint, circular transmitters. Similar to method one, we first convert the image to gray scale, blur it, and pass it through a binary OTSU filter that calculates the optimal threshold in order to further reduces the image into binary scale [70]. Then we find contours for each blob and find the minimum enclosing circle (or other shape) for each contour. After finding each of the transmitters, we examine each transmitter subregion of the image independently to decode data from each light.

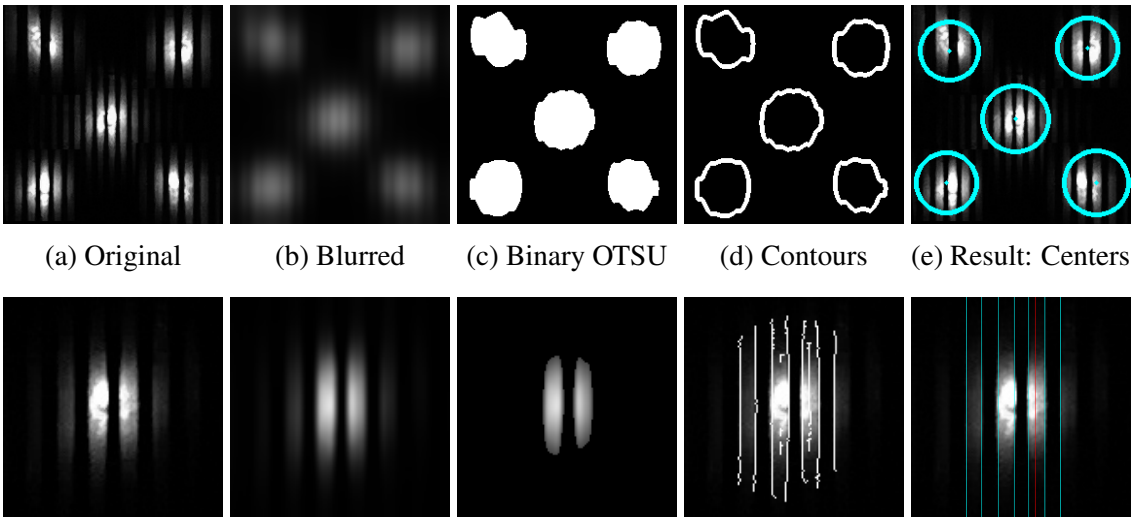
5.4.2 Recovering Frequencies

Once we find the transmitters, the data can be extracted from nearby pixels. The orientation of banding patterns only depends on the camera design and is independent of the phone's orientation. This property makes decoding easier by extracting nearby pixels along the direction that is perpendicular to the CMOS image sensor scan direction.

Fast Fourier Transform – method One. Our first method of frequency decoding samples the center row of pixels across an image subregion and takes an FFT of that vector. While this approach decodes accurately, we find that it is not very precise, requiring roughly 200 Hz of separation between adjacent frequencies to reliably decode. We find in our evaluation, however, that this approach decodes more quickly and over longer distances than method two, creating a tradeoff space, and potential optimization opportunities.

Edge detection – method Two. Figures 5.8g to 5.8j show our second method, an image processing approach. We first apply a vertical blur to the subregion and then use an OTSU filter to get threshold values to pass into the Canny edge detection algorithm [14]. Note the extreme pixelation seen on the edges drawn in Figure 5.8i; these edges are only 1 pixel wide. The transmitter captured in this subregion has a radius of only 35 pixels. To manage this quantization, we exploit the noisy nature of the detected vertical edge and compute the weighted average of the edge location estimate across each row, yielding a subpixel estimation of the column containing the edge.

Near the transmitter center, erroneous edges are sometimes identified if the intensity of



(a) Original (b) Blurred (c) Binary OTSU (d) Contours (e) Result: Centers

(f) Subregion (g) Vertical Blur (h) ToZero OTSU (i) Canny Edges (j) Result

Figure 5.8: Image processing pipeline—The top row of images encompasses our transmitter locating method. The bottom row of images demonstrates our image processing-based approach to frequency recovery. The image presented here has been edited to move the transmitters closer together for presentation.

an *on* band changes too quickly. We majority vote across three rows of the subregion (the three rows equally partition the subregion) to decide if each interval is light or dark. If an edge creates two successive light intervals, it is considered an error and removed. Using these precise edge estimates and the known scan rate, we convert the interval distance in pixels to the transmitted frequency with a precision of about 50 Hz., offering roughly 120 channels (6 kHz/50 Hz). In addition to the extra edge detection and removal, we also attempt to detect and insert missing edges. We compute the interval values between each pair of edges and look for intervals that are statistical outliers. If the projected frequency from the non-outlying edges divides cleanly into the outlier interval, then we have likely identified a missing edge, and so we add it.

5.4.3 Manchester Decoding

Manchester encoding is a common modulation used in VLC. It provides balanced luminaire brightness, a potentially higher data rate than FSK, and is simple to implement on low-end devices. However, decoding an unknown symbol rate signal requires a series of matched filters at different symbol rates. This step has the potential to be the bottle neck of the decoding process. To enhance data rate, a short preamble is desired, but this may lead to falsely locking the sequence at a 50% duty cycle if the preamble is not properly designed.

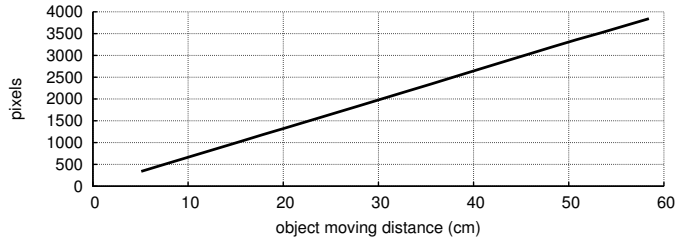


Figure 5.9: Camera distortion. An object is moved parallel to the camera from the center to the edge of the imager. We record the object's position on the imager every 2.54 cm step. The result shows that the image is highly linear even along the edge of imager.

One challenge of decoding on a camera is correctly estimating the symbol rate. This is due to discrete pixels. It becomes worse when increasing the symbol rate. Also, the transmitter and camera are not time synchronized, which results in a few pixel difference on the width of the bands. Hence, the discrete pixel effect could be alleviated by using a symbol rate that is aliquot of the rolling shutter scan rate.

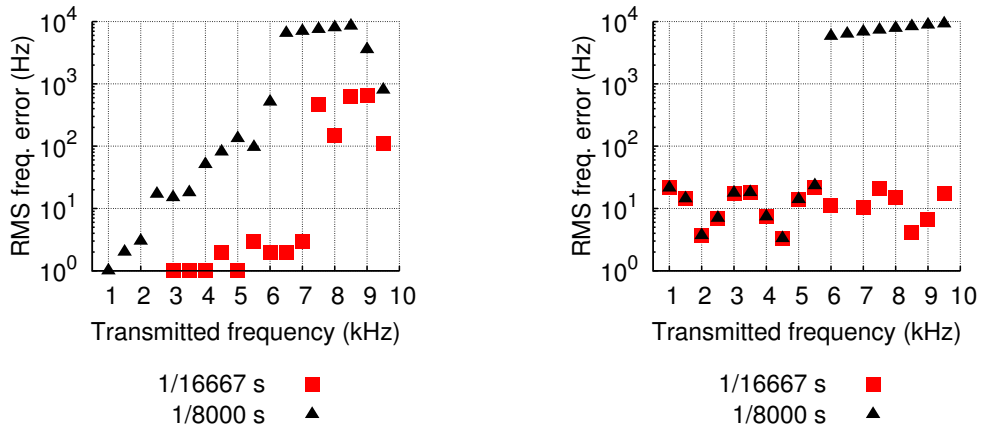
Another alternative Manchester encoding is to use a 2X symbol rate pure tone as an idle pattern between packets. The main advantage is to quickly determine the symbol rate/frequency of the idle pattern using an FFT. This approach is based on a frequency null of the Manchester encoded signal at 2X its frequency. This method requires that a fair portion of the idle pattern is captured in the image, which reduces the overall utilization.

5.5 Evaluation

In this section, we evaluate the camera communication primarily on decoding the FSK and Manchester encoding signal using single frame.

5.5.1 Lens Characterization

Due to the non-linearity of the camera lens, the captured image often has distortion and vignetting, which is more severe around the corner of the lens. To alleviate the distortion, most smartphone cameras digitally correct for it in the camera firmware [76]. The Lumia 1020 on the other hand, only utilizes the center of the optics to minimize the distortion [67]. To verify the presence and quality of distortion correction in the Lumia, we move an object from the center to the edge of the camera's frame. We find that the Lumia's images show very little distortion, deviating at most 3 pixels from the expected location as shown in Figure 5.9.



(a) Using edge detection

(b) Using FFT

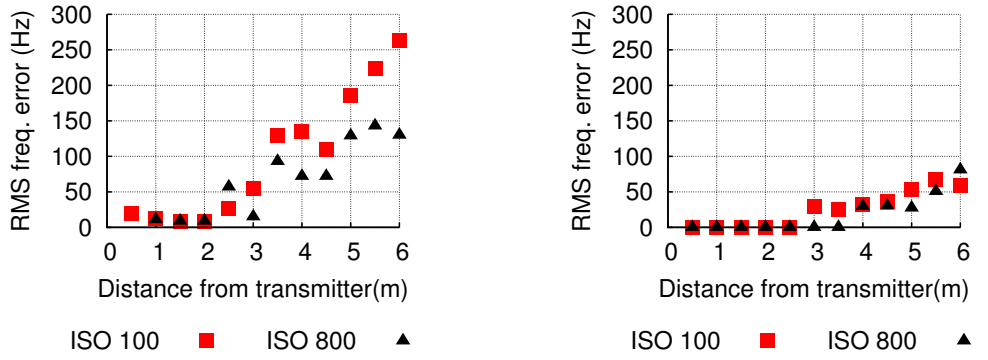
Figure 5.10: Frequency recovery at 0.2 m, 1/16667 s, 1/8000 s, and ISO 100. The edge detector performs better until ~7 kHz when quantization causes it to fail completely. The FFT method has lower resolution but can decode a wider frequency range.

5.5.2 Frequency Recovery

We evaluate two frequency decoders and find that the FFT is more robust, but that edge-detection gives better results when it succeeds.

Rx Frequency Error vs Tx Frequency. We place the camera close to the transmitter to capture a large image and sweep the transmit frequency from 1 to 10 khz in 500 Hz steps. The process is repeated using two exposure settings: 1/8000 s and 1/16667 s. Figure 5.10 shows the frequency recovery capability of the two methods. Image contrast of long exposures is lower, so frequency decodability is worse at higher transmit frequencies. For short exposure images, edge detection performs well at low transmit frequencies. The RMS frequency error at low frequencies is less than 10 Hz. In contrast, FFT provides a frequency error around 30 Hz but is more robust at higher transmit frequencies. The result suggests that a minimum 100 Hz frequency separation for FFT is required.

Rx Frequency Error vs Tx Distance. As the distance between the transmitter and phone increases, the received energy at each pixel drops due to line of sight path loss [22]. The area of the transmitter projected onto the imager plane also decreases. These factors reduce the ability to decode information. In Figure 5.11 we use a 10 cm diameter 14 W Commercial Electric can light to explore the impact of distance on our ability to recover frequency as well as the effect of varying the ISO to attempt to compensate for the lower received power. As intensity fades, the edge detection cannot reliably detect edges and it fails. The FFT method is more robust to this failure, as it is able to better take advantage of pixels with medium intensity.



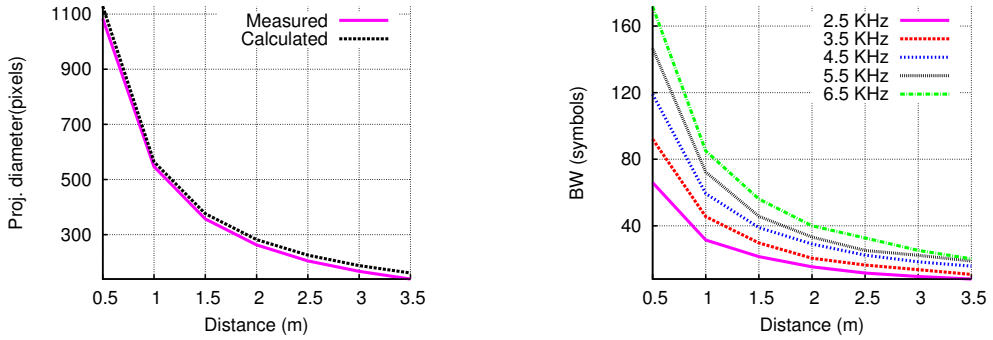
(a) Using edge detection

(b) Using FFT

Figure 5.11: As distance grows, the light intensity and area fall superlinearly. Using a higher ISO amplifies what little light is captured, enhancing frequency recoverability. We transmit a 1 kHz frequency on a commercial LED and find that the decoded frequency error remains under 100 Hz for distances up to 6 m from the transmitter.

The Importance of Frequency Recovery. Human and optics constraints limit our bandwidth to 1 to 10 kHz. With an effective resolution of 200 Hz, the FFT decoder can only identify about 46 symbols and each symbol could be 5-bit data.

5.5.3 Manchester Decoding



(a) Transmitter length

(b) Maximum available bandwidth

Figure 5.12: Bandwidth versus distance. As the distance grows, the size of a transmitter projected onto the imager plane becomes smaller. Hence, the maximum available bandwidth decreases. Higher transmit frequencies (shorter symbol time) can help.

Bandwidth vs Distance. As distance grows, not only is the received power reduced but also the area projected onto the image plane is shrunk. The projected area can be expressed as $\frac{X \times Z_f}{D}$, where X is the width of transmitter and D is the distance between the

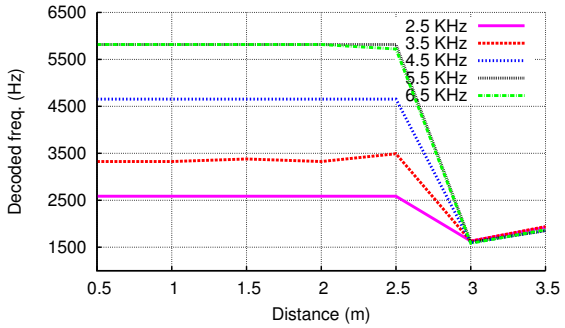
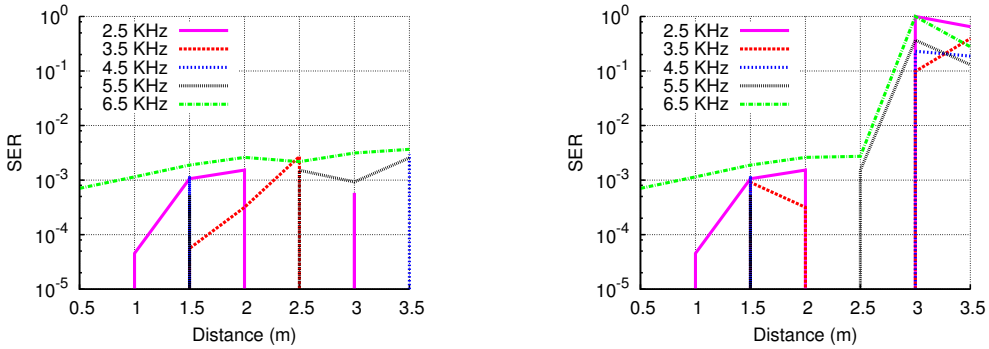


Figure 5.13: Frequency estimation using a matched filter. The discrete pixel effect results in inaccuracy in estimating high transmit frequencies. A matched filter fails to distinguish between 5.5 kHz and 6 kHz.



(a) SER (Known frequency)

(b) SER (Unknown frequency)

Figure 5.14: Examining the decodability of Manchester data across various transmit frequencies and distances. If the frequency is known to the receiver, a 4-bit symbol can be correctly decoded with high probability. However, using a matched filter to estimate transmit frequency fails above 3 m, which leads to a failure to decode the symbol.

transmitter and the camera. Figure 5.12a shows the inversely proportional relationship. At 3 m distance, the size of a 10 cm transmitter on an imager is only 83 pixels (Lumia 1020). Figure 5.12b shows that a higher transmit frequency allows for more symbols at a given distance. The maximum available bandwidth at 3 m using 6.5 kHz Manchester encoded signal is only 25 symbols.

Matched filter frequency estimation. We evaluate frequency decodability using a matched filter on a Manchester encoded signal. Figure 5.13 shows the discrete pixel effect on high transmit frequencies. A matched filter fails to distinguish 5.5 kHz and 6 kHz, and frequency estimation fails completely at 3 m.

Manchester decoding. We further explore the probability of decoding a symbol using Manchester encoding at different distances. The transmitter repeatedly sends a constant

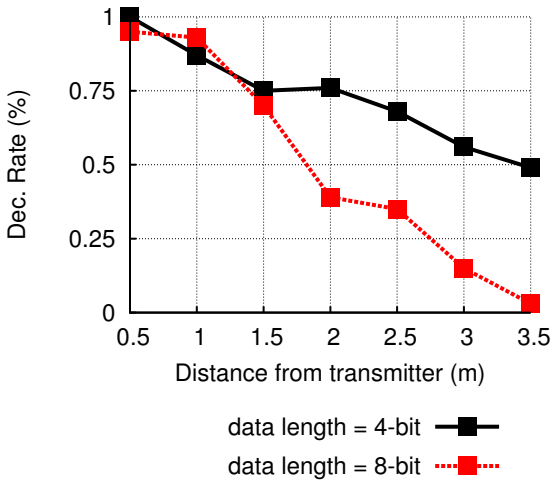


Figure 5.15: Hybrid decoding is able to better tolerate the frequency quantization ambiguity than pure Manchester. Shorter data has a higher probability of being correctly decoded at long distances.

4-bit symbol without any packet structure. Figure 5.14 indicates the symbol can be correctly decoded with a symbol error rate (SER) around 5×10^{-2} at various frequencies and distances. However, if the transmit frequency is unknown to the receiver, a matched filter is required to estimate the frequency. The frequency estimation fails above 3 m, and the symbol cannot be correctly decoded.

Manchester with idle pattern. Instead of using a matched filter to estimate frequency, we use an FFT to extract the frequency of the idle pattern. This approach requires a packet structure in order to detect the start of a packet. We encode data at 3 kHz with a 6 kHz idle pattern, and sweep across distance. Figure 5.15 shows the packet reception rate under various distances and 2 lengths of packet. Shorter packets have higher packet reception rates.

5.6 Summary

The wide spread of CMOS imagers and their rolling shutter characteristics create huge opportunities for VLC applications. The key requirements for a camera are imager resolution, film speed, and exposure setting. Imager resolution will keep improving thanks to advanced CMOS technology, and low-level control over the camera is gradually exported to the user. Several modulation, image processing, and demodulation approaches for camera communication are introduced and evaluated. For a frequency modulated signal, FFT is a robust decoding technique, achieving less than 100 Hz frequency error and a 6 m com-

munication range using a 10 cm LED transmitter. For a Manchester encoded signal, the available bandwidth on single image is small. The result suggests only a few bits (˜4) can be decoded reliably in the normal use case (˜3 m). A multi-frame approach may be used to increase the data rate. The challenge for multi-frame camera communication is synchronizing the transmitter with the data frames, as well as designing a coding scheme that works on different phones with various capabilities. Having demonstrated how it is possible to communicate data from LED luminaires to camera receivers, a technique generically known as CamComm, we next demonstrate how it is possible to build an indoor localization system on this infrastructure.

CHAPTER 6

VLC Positioning (VLCP)

Accurate indoor positioning can enable a wide range of location-based services across many sectors. Retailers, supermarkets, and shopping malls, for example, are interested in indoor positioning because it can provide improved navigation, which helps avoid unrealized sales when customers cannot find items they seek, and increases revenues from incremental sales from targeted advertising [42]. Indeed, the desire to deploy indoor location-based services is one reason that the overall demand for mobile indoor positioning in the retail sector is projected to grow to $5 billion by 2018 [19]. However, despite the strong demand forecast, indoor positioning remains a "grand challenge," and no existing system offers accurate location and orientation using unmodified smartphones [57].

WiFi and other RF-based approaches deliver accuracy measured in meters and no orientation information, making them a poor fit for many applications like retail navigation and shelf-level advertising [8, 16, 92]. Visible light-based approaches have shown some promise for indoor positioning, but recent systems offer landmarks with approximate room-level semantic localization [73], depend on custom hardware and received signal strength (RSS) techniques that are difficult to calibrate, or require phone attachments and user-in-the-loop gestures [57]. These limitations make deploying indoor positioning systems in "bring-your-own-device" environments, like retail, difficult.

Motivated by a recent claim that "the most promising method for the new VLP systems is angle of arrival" [7], we propose a new approach to accurate indoor positioning—Luxapose—that leverages Software-Defined Lighting, camera-enabled smartphones, and retailer-specific mobile applications. In our design, smart luminaires act as location beacons that encode their identities in frequency. The smartphone's camera takes pictures periodically and these pictures are processed to determine if they contain any beacons by testing for energy in a target spectrum of the columnar FFT of the image. If beacons are present, the images are decoded to determine the beacon location and identity. Once beacon identities and coordinates are determined, an angle-of-arrival localization algorithm determines the phone's absolute position and orientation in the local coordinate system.

Parameter	EZ [16]	Radar [8]	Horus [92]	Epsilon [57]	Luxapose [50]
Accuracy	2-7 m	3-5 m	∼1 m	∼0.4 m	∼0.1 m
Method	Model	FP	FP	Model	AoA
Database	Yes	Yes	Yes	No	Yes
Overhead	Minimum	WD	WD	DC	DC

Table 6.1: Comparison of WiFi- and VLC-based localization. FP, WD, AoA, and DC are fingerprinting, war-driving, angle-of-arrival, and device configuration, respectively.

Our approach is largely motivated by emerging retail environments that now include solid-state lighting and connected mobile devices. Retail settings often support a high density of overhead lights, at the aisle or hallway level, over which facility owners have control. Recent trends show growing interest in deploying solid-state LED lighting that leverages existing investments in the infrastructure, leading the lighting industry to offer a myriad of options that are electrically and mechanically compatible with today's fixtures and luminaires.

Retail customers have also shown a willingness to install retailer-specific apps on their mobile devices. These apps provide rich and dynamic content including store layouts, barcode readers, product lookups, and advertisements. To ensure high quality network access to the servers that provide the content for these apps, and to compensate for potentially poor cellular coverage indoors, retailers deploy their own WiFi access points. Driven by the need for higher bandwidth and lower latency, cloudlets are also emerging as an intermediary between cloud and mobile platforms [79].

Emerging mobile platforms offer high-performance processors, dedicated GPUs for gaming and computer vision, multi-megapixel CMOS imagers with rolling shutters, front-facing "selfie" cameras for video conferencing, low-power inertial sensors for context sensing navigation, and new form factors like wearables. When coupled with smart luminaires, retailer-specific mobile apps, and local cloudlet servers, mobile platforms could offer dynamic and responsive location based services in indoor environments. But, the key enabler for many of these services is accurate indoor positioning, which remains a "grand challenge" today [57].

Envisioning a future in which these trends merge, we explore the viability and performance of indoor positioning under conditions that are typical of retail and enterprise environments. Our results, using unmodified mobile phones and Software-Defined Lighting, offers best-in-class location (decimeter-level) and orientation error (3°), as shown in Table 6.1. Although it is difficult to directly compare different systems, we adopt the parameters proposed by Epsilon [57] for this comparison.

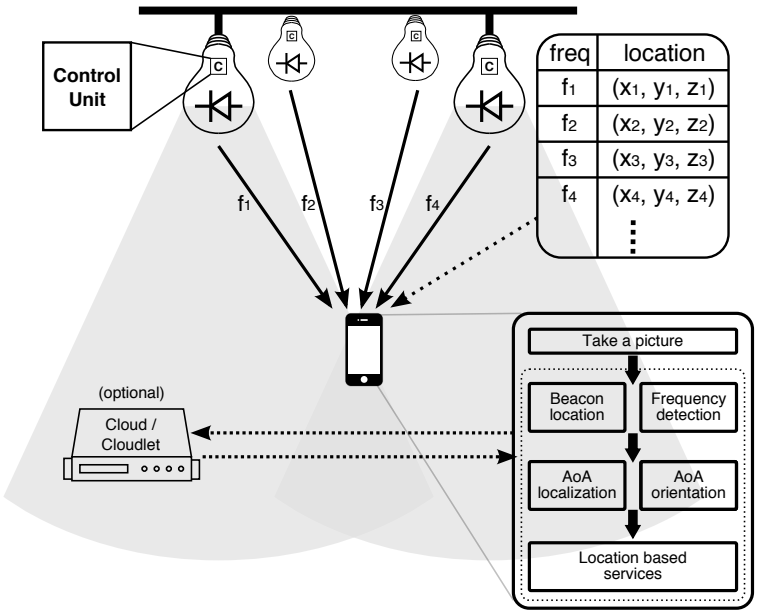


Figure 6.1: Indoor positioning system architecture. The system consists of visible light beacons, mobile phones, and a cloud/cloudlet server. Beacons transmit their identities or coordinates using human-imperceptible visible light. A phone receives these transmissions using its camera and recruits a combination of local and cloud resources to determine its precise location and orientation relative to the beacons' coordinate system using an angle-of-arrival localization algorithm. Once its position is known, a phone can support location-based services in concert with the cloud.

6.1 System Overview

The Luxapose indoor positioning system consists of visible light beacons, smartphones, and a cloud/cloudlet server, as Figure 6.1 shows. These elements work together to determine a smartphone's location and orientation, and support location-based services. Visible light beacons are smart luminaires broadcasting their identity and/or coordinates. The front-facing camera in a hand-held smartphone takes pictures periodically. These pictures are processed to determine if they contain beacons by testing for the presence of certain frequencies. If beacons are likely present, the images are decoded to both determine the beacon locations in the image itself and also to extract data encoded in the beacons' modulated transmissions. A lookup table may be consulted to convert beacon identities into corresponding coordinates if these data are not transmitted. Once beacon identities and coordinates are determined, an angle-of-arrival localization algorithm determines the phone's position and orientation in the venue's coordinate system. This data can then be used for a range of location-based services. Cloud or cloudlet resources may be used to assist with image processing, coordinate lookup, database lookups, indoor navigation, dynamic adver-

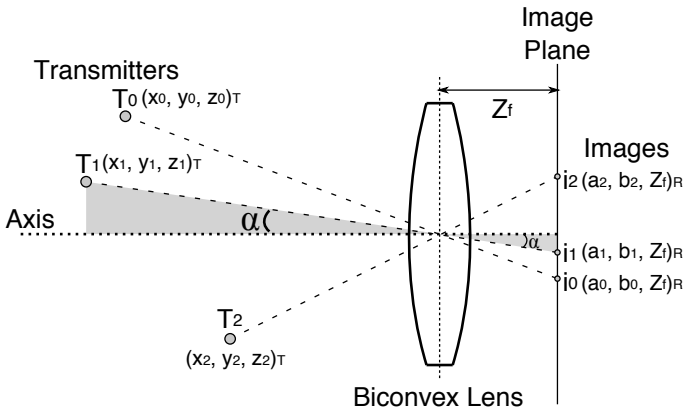


Figure 6.2: Optical AoA localization. When the scene is in focus, transmitters are distinctly projected onto the image plane. Knowing the transmitters' locations $T_j(x_j, y_j, z_j)_T$ in a global reference frame, and their image $i_j(a_j, b_j, Z_f)_R$ in the receiver's reference frame, allows us to estimate the receiver's global location and orientation.

tisements, or other services that require distributed resources.

6.2 Positioning Principles

This section presents our positioning algorithms. Our goal is to estimate the location and orientation of a mobile device assuming that we know bearings to three or more landmarks with known positions.

6.2.1 Problem Formulation

We assume that three or more point-sources (interchangeably called *beacons*, *landmarks*, and *transmitters*) have well-known 3-D coordinates. These landmarks are visible and distinguishable from each other using a mobile device's built-in camera (or *receiver*). The camera is in focus so these point sources uniquely project onto the camera imager at distinct pixel locations. Assuming that the camera geometry (e.g. pixel size, focal length, etc.) is known and the pixels onto which the landmarks are projected can be determined, we seek to estimate the position and orientation of the mobile device with respect to the landmarks' coordinate system. This problem is a variation on the well-known bearings-only robot localization and mapping problem [25].

6.2.2 Optical Angle of Arrival Localization

Luxapose uses optical angle-of-arrival (AoA) localization principles based on an ideal camera with a biconvex lens. An important property of a simple biconvex lens is that a ray of light that passes through the center of the lens is not refracted, as shown in Figure 6.2. Thus, a transmitter, the center of the lens, and the projection of transmitter onto the camera imager plane all form a straight line. Assume that transmitter T_0, with coordinates $(x_0, y_0, z_0)_T$ in the transmitters' global frame of reference, has an image i_0, with coordinates $(a_0, b_0, Z_f)_R$ in the receiver's frame of reference (with the origin located at the center of the lens). T_0's position falls on the line that passes through $(0, 0, 0)_R$ and $(a_0, b_0, Z_f)_R$, where Z_f is the distance from lens to imager in pixels. By the geometry of similar triangles, we can define an unknown scaling factor K_0 for transmitter T_0, and describe T_0's location $(u_0, v_0, w_0)_R$ in the receiver's frame of reference as:

$$
\begin{aligned}
u_0 &= K_0 \times a_0 \\
v_0 &= K_0 \times b_0 \\
w_0 &= K_0 \times Z_f
\end{aligned}
$$

Our positioning algorithm assumes that transmitter locations are known. This allows us to express the pairwise distance between transmitters in both the transmitters' and receiver's frames of reference. Equating the expressions in the two different domains yields a set of quadratic equations in which the only remaining unknowns are the scaling factors K_0, K_1, $\ldots$, K_n. For example, assume three transmitters T_0, T_1, and T_2 are at locations $(x_0, y_0, z_0)_T$, $(x_1, y_1, z_1)_T$, and $(x_2, y_2, z_2)_T$, respectively. The pairwise distance squared between T_0 and T_1, denoted $d^2_{0,1}$, can be expressed in both domains, and equated as follows:

$$
\begin{aligned}
d^2_{0,1} &= (u_0 - u_1)^2 + (v_0 - v_1)^2 + (w_0 - w_1)^2 \\
&= (K_0 a_0 - K_1 a_1)^2 + (K_0 b_0 - K_1 b_1)^2 + Z_f^2 (K_0 - K_1)^2 \\
&= K_0^2 \left|\overrightarrow{Oi_0}\right|^2 + K_1^2 \left|\overrightarrow{Oi_1}\right|^2 - 2K_0 K_1 (\overrightarrow{Oi_0} \cdot \overrightarrow{Oi_1}) \\
&= (x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2,
\end{aligned}
$$

where $\overrightarrow{Oi_0}$ and $\overrightarrow{Oi_1}$ are the vectors from the center of the lens to the images i_0 (a_0, b_0, Z_f) and i_1 (a_1, b_1, Z_f), respectively. The only unknowns are K_0 and K_1. Three transmitters would yield three quadratic equations in three unknown variables, allowing us to find K_0, K_1, and K_2, and compute the transmitters' locations in the receiver's frame of reference.

6.2.3 Estimating Receiver Position

In the previous section, we show how the transmitters' locations in the receiver's frame of reference can be calculated. In practice, imperfections in the optics and inaccuracies in estimating the transmitters' image locations make closed-form solutions unrealistic. To address these issues, and to leverage additional transmitters beyond the minimum needed, we look at position estimation as an optimization problem that seeks the minimum mean square error (MMSE) over a set of scaling factors, as follows:

$$\sum_{m=1}^{N-1}\sum_{n=m+1}^{N}\{K_m^2\left|\overrightarrow{Oi_m}\right|^2+K_n^2\left|\overrightarrow{Oi_n}\right|^2-2K_mK_n(\overrightarrow{Oi_m}\cdot\overrightarrow{Oi_n})-d_{mn}^2\}^2,$$

where N is the number of transmitters projected onto the image and $\binom{N}{2}$ in the number of equations.

Once all the scaling factors are estimated, the transmitters' locations can be determined in the receiver's frame of reference, and the distances between the receiver and transmitters can be calculated. The relationship between the two domains can be expressed as follows:

$$\begin{bmatrix} x_0 & x_1 & \dots & x_{N-1} \\ y_0 & y_1 & \dots & y_{N-1} \\ z_0 & y_1 & \dots & z_{N-1} \end{bmatrix} = \mathbf{R}\times\begin{bmatrix} u_0 & u_1 & \dots & u_{N-1} \\ v_0 & v_1 & \dots & v_{N-1} \\ w_0 & w_1 & \dots & w_{N-1} \end{bmatrix}+\mathbf{T},$$

where **R** is a 3-by-3 rotation matrix and **T** is a 3-by-1 translation matrix.

The three elements of **T** (T_x, T_y, T_z) represent the receiver's location in the transmitters' frame of reference. We determine the translation matrix based on geometric relationships. Since all the scaling factors are now known, equivalent distances in both domains allow us to obtain the receiver's location in the transmitters' coordinate system:

$$(T_x-x_m)^2+(T_y-y_m)^2+(T_z-z_m)^2=K_m^2(a_m^2+b_m^2+Z_f^2),$$

where (x_m, y_m, z_m) are the coordinates of the m-th transmitter in the transmitters' frame of reference, and (a_m, b_m) is the projection of the m-th transmitter onto the image plane. Finally, we estimate the receiver's location by finding the set (T_x, T_y, T_z) that minimizes:

$$\sum_{m=1}^{N}\{(T_x-x_m)^2+(T_y-y_m)^2+(T_z-z_m)^2-K_m^2(a_m^2+b_m^2+Z_f^2)\}^2$$

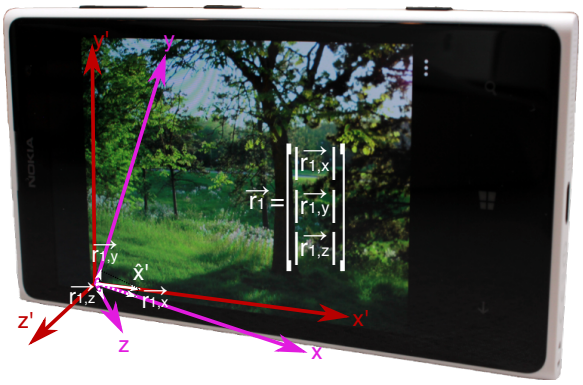


Figure 6.3: Receiver orientation. The vectors x', y', and z' are defined as shown in the picture. The projection of the unit vectors $\hat{x}'$, $\hat{y}'$, and $\hat{z}'$ onto the x, y, and z axes in the transmitters' frame of reference gives the elements of the rotation matrix **R**.

6.2.4 Estimating Receiver Orientation

Once the translation matrix **T** is known, we can find the rotation matrix **R** by individually finding each element in it. The 3-by-3 rotation matrix **R** is represented using three column vectors, $\overrightarrow{r_1}$, $\overrightarrow{r_2}$, and $\overrightarrow{r_3}$, as follows:

$$\mathbf{R} = \begin{bmatrix} \overrightarrow{r_1} & \overrightarrow{r_2} & \overrightarrow{r_3} \end{bmatrix},$$

where the column vectors $\overrightarrow{r_1}$, $\overrightarrow{r_2}$ and $\overrightarrow{r_3}$ are the components of the unit vectors $\hat{x}'$, $\hat{y}'$, and $\hat{z}'$, respectively, projected onto the x, y, and z axes in the transmitters' frame of reference. Figure 6.3 illustrates the relationships between these various vectors. Once the orientation of the receiver is known, determining its bearing requires adjusting for portrait or landscape mode usage, and computing the projection onto the xy-plane.

6.3 Location Beacons

We use smart luminaires to transmit OOK-modulated beacons to CMOS cameras. The beacons are modulated with a square wave or Manchester-encoded data stream at frequencies above 1 kHz to avoid direct or indirect flicker [83]. The beacons are received by a CMOS imager in a smartphone. Although the beacon frequency greatly exceeds the camera frame rate, we leverage the CMOS imager's rolling shutter effect to capture the high frequency beaconing, and computer vision-based photogrammetry to localize the transmitters in the image and extract their identity or data.

6.3.1 Encoding Data in Beacons

Each smart luminaire transmits a single frequency or Manchester encoded signal, allowing different luminaires to be distinctly identified. We use the techniques described in §5 to decode data.

6.3.2 Decoding Data in Images

Decoding images captured by the camera involves identifying transmitter locations in the image and extracting the data sent by each transmitter, if any. In Luxapose, we use pure tone as the location beacon, and we discussed how to extract their locations and identities in an earlier chapter.

6.4 Implementation

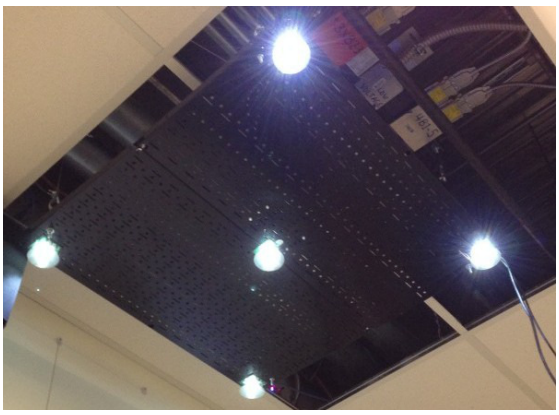
A cloudlet server implements the full image processing pipeline shown in Figure 5.8 using OpenCV 2.4.8 with Python bindings. On an unburdened MacBook Pro with a 2.7 GHz Core i7, the median processing time for the full 33 MP images captured by the Lumia is about 9 s (taking the picture: 4.46 s, upload: 3.41 s, image processing: 0.3 s, location estimation: 0.87 s) without any optimizations. The cloudlet application contains a mapping from transmitter frequency to absolute transmitter position in space. Using this mapping and the information from the image processing, we implement the techniques described in §6.2 using the `leastsq` implementation from SciPy. Our complete cloudlet application is 722 Python SLOC.

6.5 Evaluation

In this section, we evaluate position and orientation accuracy in both typical usage conditions and in carefully controlled settings. We also evaluate the visible light communications channel for pure tones, Manchester-encoded data, and a hybrid of the two. Our experiments are carried out on a custom indoor positioning testbed.

6.5.1 Experimental Methodology

We integrate five smart luminaires, a smartphone, and a cloudlet server into an indoor positioning testbed, as Figure 6.4 shows. The smart luminaires are mounted on a height-adjustable pegboard and they form a 71.1×73.7 cm rectangle with a center point. A com-



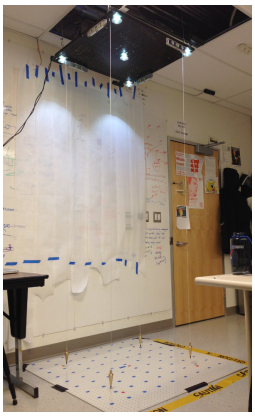


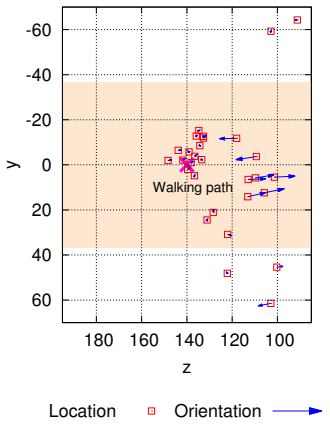
Figure 6.4: Indoor positioning testbed. Five smart luminaires beacon are mounted 246 cm above the ground for experiments. Ground truth is provided by a pegboard on the floor with 2.54 cm location resolution.

plementary pegboard is affixed to floor and aligned using a laser sight and verified with a plumb-bob, creating a 3D grid with 2.54 cm resolution of known locations for our experimental evaluation. To isolate localization from communications performance, we set the transmitters to emit pure tones in the range of 2 kHz to 4 kHz, with 500 Hz separation, which ensures reliable communications (we also test communications performance separately). Using this testbed, we evaluate indoor positioning accuracy—both location and orientation—for a person, model train, and statically.

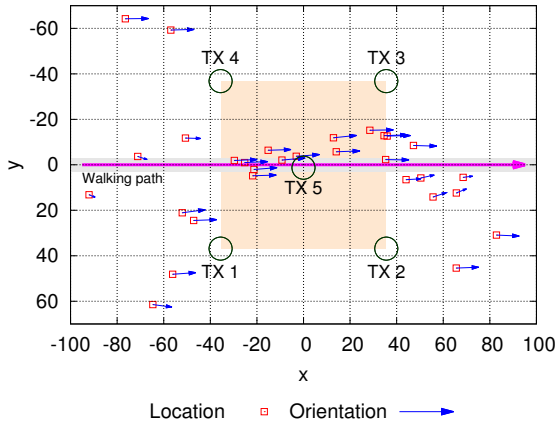
6.5.2 Realistic Positioning Performance

To evaluate the positioning accuracy of the Luxapose system under realistic usage conditions, we perform an experiment in which a person repeatedly walks under the indoor positioning testbed, from left to right at 1 $^{m}/_{s}$, as shown from the top view of the testbed in Figure 6.5b and side view in Figure 6.5d. The CDF of estimated location and orientation errors when the subject is under the landmarks (shaded) or outside the landmarks (unshaded) is shown in Figure 6.5c. When under the landmarks, our results show a median location error of 7 cm and orientation error of 6°, substantially better than when outside the landmarks, which exhibit substantially higher magnitude (and somewhat symmetric) location and orientation errors.

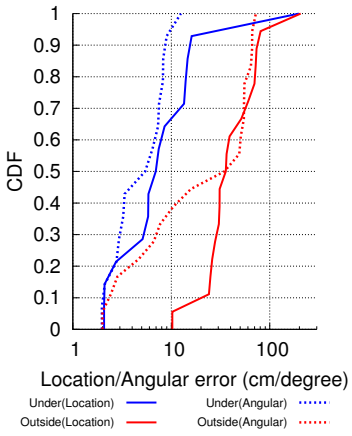
To evaluate the effect of controlled turning while under the landmarks, we place a phone on a model train running at 6.75 $^{cm}/_{s}$ in an oval, as shown in Figure 6.6a. Most of the location samples fall on or within 10 cm of the track with the notable exception of when the phone is collinear with three of the transmitters, where the error increases to about 30 cm, though this is an artifact of the localization methodology and not the motion. When the speed of



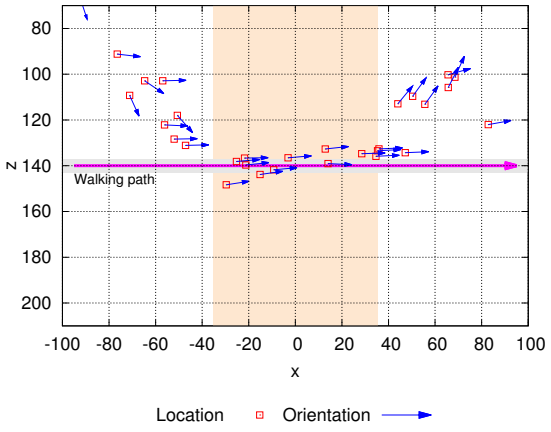
(a) YZ view (back)



(b) XY view (top)

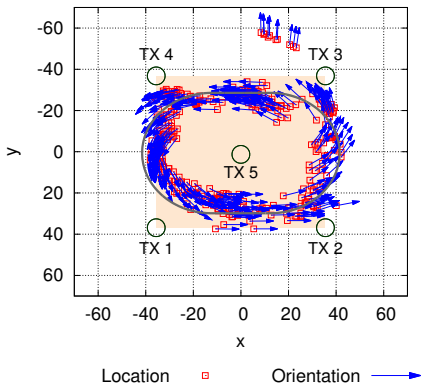


(c) Location and angular error

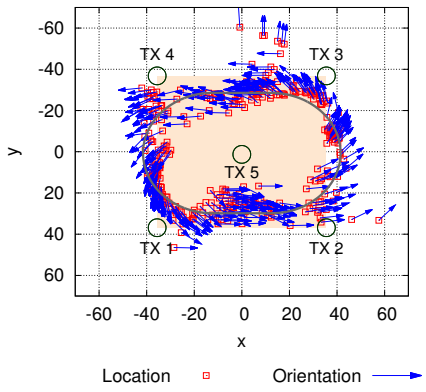


(d) XZ view (side)

Figure 6.5: Key location and orientation results under realistic usage conditions on our indoor positioning testbed. The shaded areas are directly under the lights. Figure 6.5a, Figure 6.5b, and Figure 6.5d show Luxapose's estimated location and orientation of a person walking from the back, top, and side views, respectively, while using the system. A subject carrying a phone walks underneath the testbed repeatedly, trying to remain approximately under the center ($x = -100 \ldots 100, y = 0, z = 140$). We measure the walking speed at ~1 m/s. Figure 6.5c suggests location estimates (solid line) and orientation (dotted line) under the lights (blue), have lower error than outside the lights (red).



(a) Model train moving at 6.75 cm/s



(b) Model train moving at 13.5 cm/s

Figure 6.6: Figure 6.6a and Figure 6.6b show the effect of motion blur. To estimate the impact of motion while capturing images, we place the smartphone on a model train running in an oval at two speeds. While the exact ground truth for each point is unknown, we find the majority of the estimates fall close to the track and point as expected.

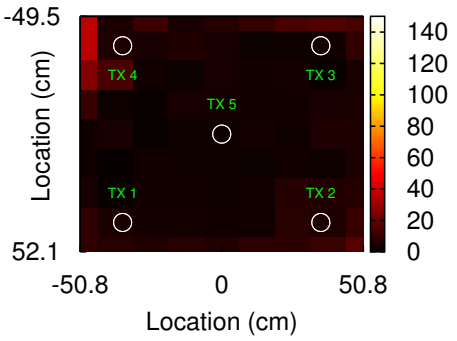
the train is doubled—to 13.5 cm/s—we find a visible increase in location and orientation errors, as shown in Figure 6.6b.

6.5.3 Controlled Positioning Accuracy

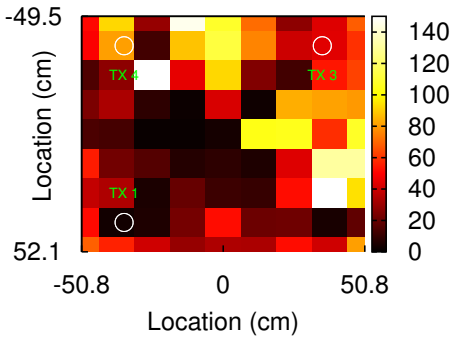
To evaluate the limits of positioning accuracy under controlled, static conditions, we take 81 pictures in a grid pattern across a 100×100 cm area 246 cm below the transmitters and perform localization. When all five transmitters are active, the average position error across all 81 locations is 7 cm, as shown in Figures 6.7a and 6.7c. Removing any one transmitter, corner or center, yields very similar results to the five-transmitter case, as seen in the CDF in Figure 6.7d.

Removing two transmitters can be done in three ways: (i) removing two opposite corners, (ii) removing two transmitters from the same side, and (iii) removing one corner and the center. Scenario (iii) introduces the largest error, captured in the heatmap in Figure 6.7b, with an average error as high as 50 cm in the corner underneath the missing transmitter. In the case of a missing side (ii), the area underneath the missing transmitters has an average error of only 29 cm. Figure 6.7d summarizes the results of removing various transmitter subsets.

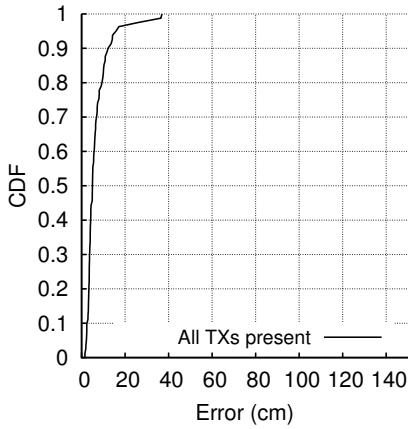
In our worst case results, on an unmodified smartphone we are able to achieve parity (˜50 cm accuracy) with the results of systems such as Epsilon [57] that require dedicated receiver hardware in addition to the infrastructure costs of a localization system. However,



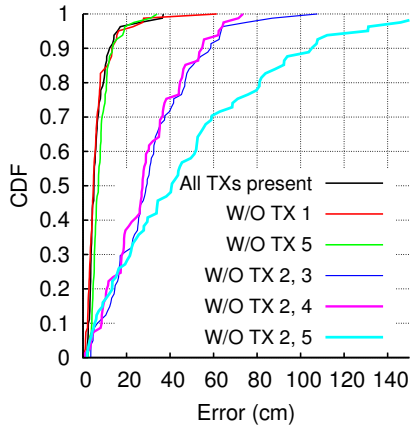
(a) Heat map with 5 TXs.



(b) Heat map W/O TX 2,5.



(c) CDF with all TXs present.



(d) CDFs when TXs removed.

Figure 6.7: Localization accuracy at a fixed height (246 cm). Figure 6.7a shows a heat map of error when all 5 transmitters are present in the image, and Figure 6.7c shows a CDF of the error. Figure 6.7d explores how the system degrades as transmitters are removed. Removing any one transmitter (corner or center) has minimal impact on location error, still remaining within 10 cm for ˜90% of locations. Removing two transmitters (leaving only the minimum number of transmitters) raises error to 20˜60 cm when corners are lost and as high as 120 cm when the center and a corner are lost. As shown in the heat map in Figure 6.7b, removing the center and corner generates the greatest errors as it creates sample points with both the largest minimum distance to any transmitter and the largest mean distance to all transmitters.

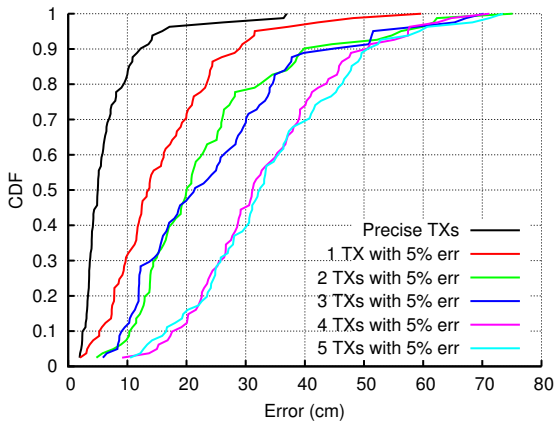


Figure 6.8: CDF of location error from a 5% error in absolute transmitter location under the same conditions as Figure 6.7a. This experiment simulates the effect of installation errors.

with only one additional transmitter in sight, we are able to achieve an order of magnitude improvement in location accuracy.

Thus far, we have assumed the precise location of each transmitter is known. Figure 6.8 explores the effect of transmitter installation error on positioning by introducing a 5% error in 1–5 transmitter positions and re-running the experiment from Figure 6.7a. With 5% error in the origin of all five transmitters, our system has only a 30 cm 50$^{\text{th}}$ percentile error, which suggests some tolerance to installation-time measurement and calibration errors.

To evaluate the orientation error from localization, we rotate the phone along the x', y', and z' axes. We compute the estimated rotation using our localization system and compare it to ground truth when the phone is placed 240 cm below the 5 transmitters. Figure 6.9 shows the orientation accuracy across all 3 rotation axes. The rotation errors fall within 3° in all measurements.

6.6 Discussion

In this section, we discuss some limitations of our current system and potential directions for future work.

Deployment Considerations. In real settings, all luminaire locations must be known, although only the relative distances between closely located luminaires must be known with high accuracy. Although not trivial, it does not seem difficult to ensure that this condition holds. We have deployed a grid of sixteen luminaires in our lab, and we analyze the effect of location errors on localization accuracy in §6.5.3. We note that almost any localization system must know its anchor locations. In a practical setting, this would be

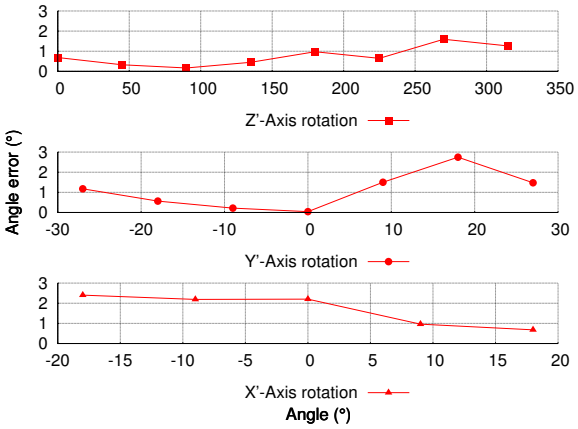


Figure 6.9: We rotate the mobile phone along axes parallel to the z'-, y'-, and x'-axis. Along the z'-axis, the mobile phone rotates 45° at a time and covers a full circle. Because of FoV constraints, the y'-axis rotation is limited to -27° to 27° and the x'-axis is limited to -18° to 18° with 9° increments. The experiments are conducted at a height of 240 cm. The angle error for all measurements falls within 3°.

done, presumably, with the aid of blueprints and a laser rangefinder.

Usability. Our system targets an active user, so that the front-facing camera naturally observes the ceiling during use. Passive localization (e.g. while the phone is in a pocket) is out of scope.

Distance. Distance is the major limitation for our system. Received signal and projected image size are strongly affected by distance. We find that a 60 pixel projection is roughly the lower bound for reliable frequency decoding. However, as camera resolutions increase, our usable distance will improve.

Local Filtering. Not all images capture enough transmitters to successfully localize. It would be desirable to perform some local filtering to discard images that would not be useful for positioning, thus avoiding the cost of transferring undecodable images to the cloud. We explore one such possibility in Figure 6.11b. The phone selects a sampling of image rows and performs an FFT, searching for the presence of high frequency components. This fast and simple algorithm rejects many images that would not have decoded properly.

Recursive Searching. The captured image can be further partitioned into smaller vertical segments. Assuming the total width of the image is W pixels and it is partitioned into X vertical segments, the FFT computational complexity on each segment is $\mathcal{O}(\frac{W}{X}log(\frac{W}{X}))$. Thus, the complexity of all X segments is $\mathcal{O}(Wlog(\frac{W}{X}))$, which suggests that smaller segments ($\frac{W}{X}$) are faster (ignoring processing overhead). However, using smaller segments to determine frequency leads to higher frequency error due to fewer FFT points and partially covered transmitters. Alternatively, another possible approach for locating transmitters in

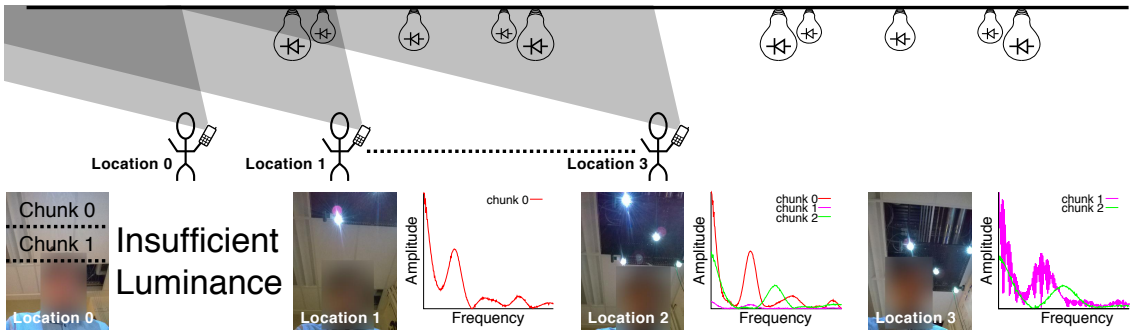
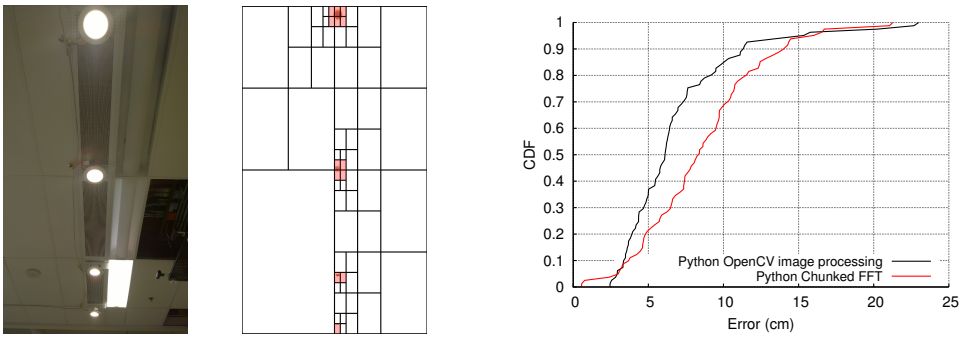


Figure 6.10: Local filtering. In this experiment, we walk under our testbed, capturing images at about 1 fps. We divide each frame into 8 "chunks" and run an FFT along the center row of pixels for each chunk. The FFTs of non-negligible chunks are presented next to each image. At each location, we also capture an image taken with traditional exposure and film speed settings to help visualize the experiment. The FFTs are performed on images captured with 1/16667 s exposure on ISO 100 film.



(a) Normal Exp. (b) Partitioned Img. (c) CDF comparison.

Figure 6.11: Recursive Searching. The image is partitioned into many horizontal segments. Each segment is quickly scanned by taking an FFT of its column sum. Segments with no peaks are discarded and segments with interesting peaks are recursed into until reaching a preset lower bound. These subimages can be sent to the cloudlet server to perform further image processing. Alternatively, the location of each transmitter could be estimated using weighted coordinates to skip the image processing. Figure 6.11c shows the performance degradation using estimated locations. In those controlled 81 samples, the location accuracy is reduced by less than 5 cm.

Figure 6.12: (left) The same LED tube imaged twice at 90° rotations shows how multiple beacons can be supported in a single lamp.
(right) A single fixture can support multiple LED drivers (four here). An image capturing only this fixture could be used to localize.

the captured image, like Figure 6.11a, may be a divide and conquer technique, as shown in Figure 6.11b. As this algorithm already partitions the image into bins with FFTs, it is also well suited to solve the problem of separating non-disjoint transmitters.

To accurately estimate location, these image segments can be uploaded to a cloudlet server to perform clustering and further image processing. This significantly reduces the traffic sent to the cloud/cloudlet—from 33 MP to 0.42 MP (13 chunks $\times$ (33/1024) MP/chunk), dramatically reducing transfer time and the processing time on the cloudlet. This approach may even allow positioning to occur entirely on the smartphone. Moreover, the image processing pipeline could be entirely skipped by estimating transmitters' locations using weighted centers. The location can be estimated using a linear combination of clustered segments' centroids with FFT power as weights. Figure 6.11c demonstrates accuracy degradation using estimated location in a controlled setting. It suggests that accuracy would only degrade by 5 cm while the image processing could be sped up by 15%.

Figure 6.12 shows how a single LED tube can transmit multiple beacons (left) and how a fixture with multiple tubes could support the non-collinear transmitter requirement (right). Localizing with this fixture would require improving our image processing, which currently assumes disjoint, circular transmitters.

Interference. Since only the direct line-of-sight path is captured by our short exposure time, there is little danger from interference regardless of transmitter density (for two transmitters' projections to alias, the pixel quantization must be so poor that they are only mapping to a few pixels and are undecodable anyway).

Limited Frequency Channels. From the frequency recovery result we showed in

§5.5.2, our system has a limited set (up to 46) of frequencies with which to label each transmitter. One method to increase the number of labels would be to have each transmitter alternate between two frequencies ($\binom{46}{2} = 1035$). Reliably and accurately estimating inter-frame motion (e.g. using the accelerometer and gyroscope), however, could prove difficult, making it difficult to match transmitter projections across frames.

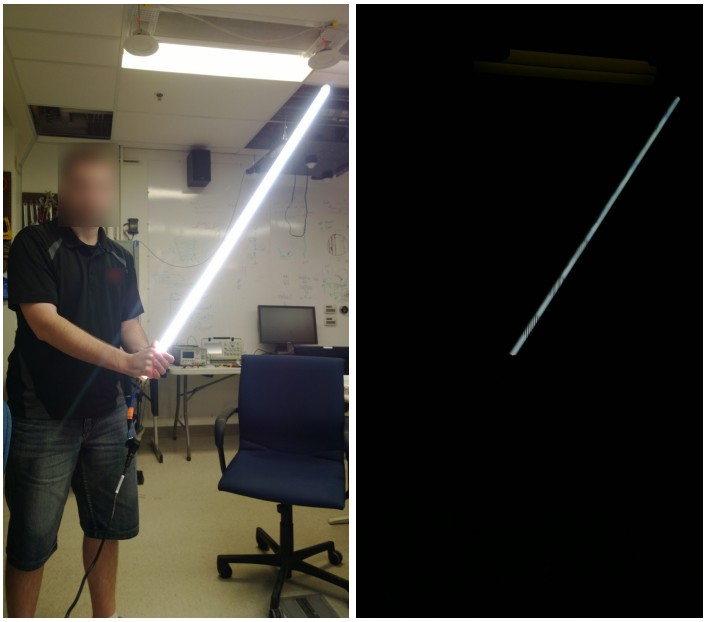
A simpler approach that still requires only a single image is to simply re-use labels and leverage transmitter adjacency relationships. As our system captures contiguous images and requires at least three landmarks to localize, the adjacency relationships between lights form another constraint that can uniquely identify transmitters. Actually identifying transmitters with this system is surprisingly simple. For each frequency observed, consider all possible transmitter locations and compute the total inter-transmitter distance. The set of transmitters that minimizes this distance are the actual transmitters. This transmitter labeling technique is the same minimization procedure already used by the processing for AoA estimation.

Dimmable LEDs. Dimming is a fundamental requirement in IEEE 802.15.7 [5]. LEDs can be dimmed by either reducing their current or using PWM. In Luxapose design, direct PWM dimming does not affect location beacons (frequencies).

Privacy. Our design does not require interaction with the local environment. Luminaires are unidirectional beacons and image capture emits no signals. If needed, the lookup table can be acquired once out of band, and processing could be done either on the phone or a user's private cloud. A user can thus acquire location estimates without sharing any location information with any other entity. Also, the camera is set to an extremely low exposure rate. Under normal operation, the captured image doesn't not contain any facial information. Figure 6.13 shows only bright region (LED tube) can be seen on the extremely short exposure image.

6.7 Summary

Accurate indoor positioning has been called a "grand challenge" for computing. In this chapter, we take a small step toward addressing this challenge by showing that software-defined lighting can support accurate indoor positioning with higher accuracy than prior work. Our results show that it is possible to achieve decimeter location error and 3° orientation error by simply walking under overhead lights while using one's smartphone. When used in typical retail settings with overhead lighting, this allows a user to be accurately localized every few meters, perhaps with dead reckoning filling in the gaps. Although our current approach has many drawbacks, none appear to be fundamental. Having demon-



(a) Normal exposure (b) Short exposure

Figure 6.13: Luxapose captures image using extremely short exposure time. Thus, users' facial information will not be captured.

strated the key building blocks of a software-defined lighting system—transmitters, receivers, modulation schemes, and programming interfaces—we now turn our attention to the problem of unifying these various components into a single, cohesive system because, ultimately, there is only one lighting infrastructure that must be shared in order to support these envisioned applications in a cost effective manner.

CHAPTER 7

Architectural Evaluation

In this chapter, we evaluate our central thesis: that solid-state lighting enables a diverse array of post-illumination applications that can be simultaneously realized by a shared lighting infrastructure, under software-defined communications and control, without adversely affecting the primary illumination goal of indoor lighting.[1] To do so, we construct a software-defined lighting testbed that demonstrates the key architectural principles, incorporates the various hardware and software elements of the architecture, and simultaneously supports four different sample applications that are expressed using our SDL API.

7.1 Prototype Software-Defined Lighting Testbed

Figure 7.1 shows our prototype software-defined lighting testbed. Our testbed incorporates most of our architectural elements including: (i) commercial and custom luminaires; (ii) Ethernet, powerline, and wireless communication backhauls; (iii) a software-defined lighting controller; and (iv) several software-defined lighting applications.

7.1.1 Luminaires

We employ several different kinds of luminaires in our testbed. The first type of luminaire employs a custom LED frontend board, a custom control plane using an 802.15.4 radio, and a powerline communications (PLC) data plane. The second type of luminaire employs a commercial LED frontend, an Actel FPGA-based control plane that employs our Floodcasting wireless primitive for synchronization, and an Ethernet-based data plane. The third type of luminaire employs a custom LED frontend, and custom control and data planes that use an 802.15.4 radio. Finally, a fourth type of luminaire employs either a custom

[1] Without loss of generality, we implement some applications that demonstrate interoperability with pre-existing commercial products and research artifacts, but which are perceptible by humans due to their low modulation rates.

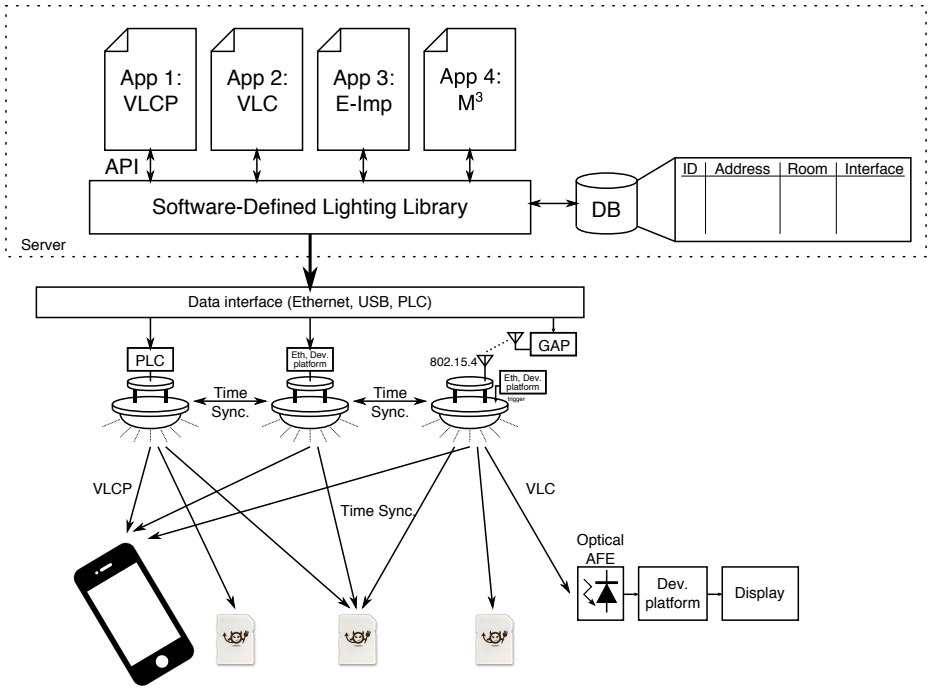


Figure 7.1: Prototype software-defined lighting testbed.

or commercial LED frontend but a fixed-function control/data plane that simply oscillates the LED at a DIP-switch configurable rate. Collectively, these luminaires demonstrate a multitude of frontends, control planes, and data planes, demonstrating the generality of our approach.

7.1.2 Communication Backhauls

The communication backhauls over which the data plane runs is an important design point. Some backhauls, like 802.15.4, are relatively slow (250 kb/s) but easy to deploy, while others, like Ethernet, are fast (10–1000 Mb/s) but difficult route to existing lighting infrastructure. PLC strikes a balance between these two extremes. PLC offers higher data rates (up to 600 Mb/s) than 802.15.4 and lower installation costs than Ethernet. To support a range of different applications, our testbed supports all three of these backhauls—802.15.4, Ethernet, and PLC.

7.1.3 Software-Defined Lighting Controller

SDL applications access the lighting infrastructure through a server called the SDL controller. This server exports an application programming interface (API) to client applications that are accessible via an SDL software library that implements the interfaces pre-

sented earlier in this dissertation (§3.1.1.2). SDL applications can access a database that includes an inventory of LED luminaires, their locations, their physical interfaces, and their addresses, all through the SDL server and software library.

7.1.4 Software-Defined Lighting Applications

Our testbed supports multiple concurrent applications. We currently achieve spectrum sharing by requiring that different applications transmit their data in one or more distinct time slots within a single, repeating super-frame structure. This approach allows applications to co-exist and share the lighting infrastructure through a combination of time and frequency division multiplexing. Richer techniques for spectrum sharing could allow for concurrent transmissions through signal superposition, but we leave exploration of such techniques to future work. We describe our applications in greater detail in the next section.

7.2 Sample Applications

To demonstrate our central claim, we simultaneously run four different SDL applications on our testbed including visible light communications based positioning (VLCP), visible light communications based messaging (VLC), time synchronization of multiple low-power devices distributed across a space, and optical programming of M3 smart dust sensors. The remainder of this section describes the applications in detail and presents the application code.

7.2.1 Visible Light Communication Positioning

This application transmits unique frequencies as the idle pattern from the luminaires contained within a single room. The application assumes that a database of luminaire entries are available, indexed by location or other query-able attributes. The application also can configure all of the luminaires to transmit a specific Bluetooth Low-Energy (BLE) advertisement beacon (if they support this feature). Since only a finite number of frequencies can be decoded using a camera receiver, the BLE beacons provide a namespace for the room- or area-level clusters of luminaires, allowing transmit frequencies to be reused in different clusters. This application supports indoor positioning using unmodified smartphones and slightly modified luminaires as described in §6.

```
/* App 1: Visible Light Communications and Positioning */
int main(){
  luminaire_list_t target_luminaires = db_get_luminaires_from_room(target_room);
  sdl_cfg_packet_t config_packet;
  config_packet.idlePattern_dutyCycle = 50;
  int idle_freq;
  luminaire_t** myLuminaire_ptr;
  for (int i=0; i<target_luminaires.numOfLuminaires; i++){
    myLuminaire_ptr = target_luminaires.luminaire_ptr;
    idle_freq = db_get_vlcp_freq(myLuminaire_ptr[i]->room, myLuminaire_ptr[i]->id);
    config_packet.idlePattern_freq = idle_freq;
    sdl_cfg_packet_transmit(config_packet, myLuminaire_ptr[i]->address);
  }
  return 0;
}
```

7.2.2 Visible Light Communications

This application demonstrates the optical equivalent of the ubiquitous “Hello World” app. It transmits a Manchester-encoded message consisting of these two words from all of the luminaires in a room. A custom receiver with a photodiode frontend and FPGA processing can extract these transmissions.

```
/* App 2: Visible Light Communications */
int main(){
  luminaire_list_t target_luminaires = db_get_luminaire_from_room(target_room);
  unsigned char myString[] = "Hello World";
  sdl_data_packet_t data_packet;
  data_packet.data_ptr = myString;
  data_packet.data_length = strlen(myString) + DATA_PKT_LEN_BASE;
  data_packet.symbol_dutyCycle = 50;
  data_packet.symbol_rate = 400000;
  data_packet.data_encoding = MANCHESTER;
  data_packet.priority = 0;
  data_packet.repeatInterval = 0;
  data_packet.numOfRepeat = 0;
  data_packet.timeToTransmit = 0;
  luminaire_t** myLuminaire_ptr;
  for (int i=0; i<target_luminaires.numOfLuminaires; i++){
    myLuminaire_ptr = target_luminaires.luminaire_ptr;
    sdl_data_packet_transmit(data_packet, myLuminaire_ptr[i]->address);
  }
  return 0;
}
```

7.2.3 Time Synchronization

This application transmits a programming and synchronization sequence optically to multiple Electric Imp boards, a commercial product whose optical programming protocol we reverse-engineered. The optical transmissions synchronize the Electric Imps, demonstrating the potential of using LED lighting to solve the low-power synchronization problem—one of the most vexing challenges for ultra low-power wireless communications.

```
/* App 3: Time Synchronization */
int main(){
  luminaire_list_t target_luminaires = db_get_luminaire_from_room(target_room);
  unsigned char eImp_progSequence[] = {0xaa, 0xaa, 0xaa, 0xaa, 0xaa, 0xaa,
                    0xaa, 0xaa, 0x2a, 0x02, 0x07, 0x00, 0x1d, 0x41, 0x00};
  sdl_data_packet_t data_packet;
  data_packet.data_ptr = eImp_progSequence;
  data_packet.data_length = 15;
  data_packet.symbol_rate = 100;
  data_packet.data_encoding = OOK;
  data_packet.priority = 0;
  data_packet.repeatInterval = 0;
  data_packet.numOfRepeat = 0;
  data_packet.timeToTransmit = 0;
  luminaire_t** myLuminaire_ptr;
  for (int i=0; i<target_luminaires.numOfLuminaires; i++){
    myLuminaire_ptr = target_luminaires.luminaire_ptr;
    sdl_data_packet_transmit(data_packet, myLuminaire_ptr[i]->address);
  }
  return 0;
}
```

7.2.4 Smart Dust Optical Programming

This application transmits the contents of a program binary for an ARM Cortex-M0 processor. The application conforms to the University of Michigan's M3 GOC protocol, used to program "Smart Dust" sensors.

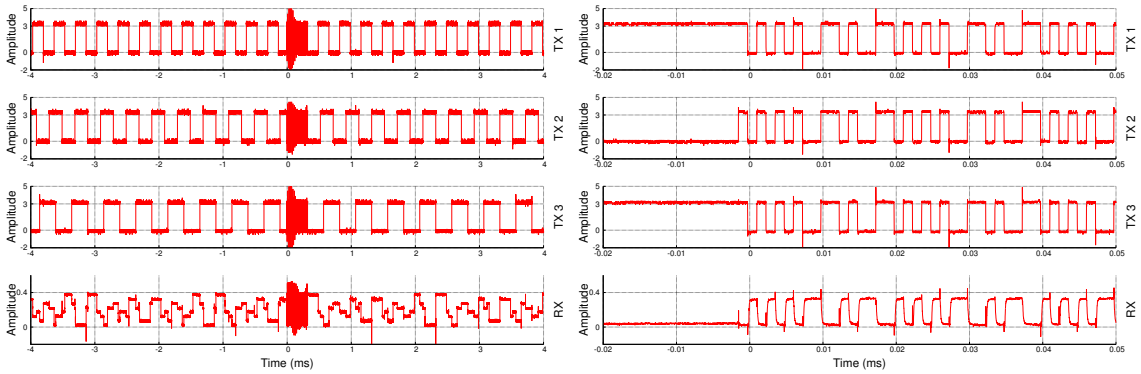
```
/* App 4: Smart Dust Optical Programming */
int main(){
  luminaire_list_t target_luminaires = db_get_luminaire_from_room(target_room);
  sdl_data_packet_t data_packet;
  data_packet.data_ptr = read_from_file(prog_file, &data_packet.data_length);
  data_packet.symbol_rate = 100;
  data_packet.data_encoding = OOK;
  data_packet.priority = 0;
  data_packet.repeatInterval = 0;
  data_packet.numOfRepeat = 0;
  data_packet.timeToTransmit = 0;
  luminaire_t** myLuminaire_ptr;
  for (int i=0; i<target_luminaires.numOfLuminaires; i++){
    myLuminaire_ptr = target_luminaires.luminaire_ptr;
    sdl_data_packet_transmit(data_packet, myLuminaire_ptr[i]->address);
  }
  return 0;
}
```

7.3 Concurrent Applications on a Shared Infrastructure

In this section, we demonstrate concurrent applications running on a shared infrastructure. These applications employ different baseband waveforms, and these waveforms use different physical layer symbols that are spectrally separated so they do not directly interfere. In addition, the data from different applications is transmitted in time-divided slots. Figure 7.3 shows a repeating super-frame structure that is used for these four applications. This figure shows three different idle patterns on three different luminaires, and the superposition of these waveforms at a single receiver.

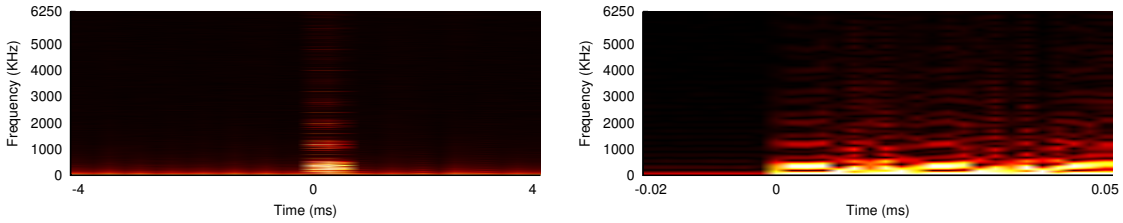
7.3.1 API Support

In this section, we discuss the generality of proposed API in §3 and potential issues of scheduling concurrent applications. The API is designed for supporting a wide range of applications, so no application specific format is included. Users are responsible for generating the entire raw optical packet (preamble, SFD, payload, CRC, etc. if necessary) before sending it to the server. By separating the application specific format, the system is capable of generating arbitrary sequences for the supported modulation schemes, enabling a wide range of applications. Improper configuration, however, can cause visible flicker, which may induce undesirable human biological responses [91]. This can be mitigated by hardware fail-safes that limit the upper bound of symbol period or an additional API to check if the waveform is DC-balanced.



(a) Localization and communication.

(b) Detail view of a communication sequence.



(c) Spectrogram.

(d) Spectrogram.

Figure 7.2: Super-frame structure showing the time and frequency division multiple access of the lighting infrastructure. In Figure 7.2a, three transmitters (top three rows) provide both localization and data communication services. Localization requires the transmitters to transmit unique idle patterns (three different frequencies) to CMOS imagers while data communication requires them to transmit identical data with symbol-level synchronization to avoid interference. In Figure 7.2b we show a zoomed-in view of data communication transmission from the same three transmitters as captured by a diffusing receiver. Tightly synchronized transmission allows diffusing receivers like photodiodes to receive data without inter-symbol interference from adjacent transmitters. Figure 7.2c and Figure 7.2d show the corresponding received spectrogram. Brighter regions represent stronger intensity. Notice increase in high-frequency energy during the concurrent transmission that is absent in the unsynchronized idle patterns.

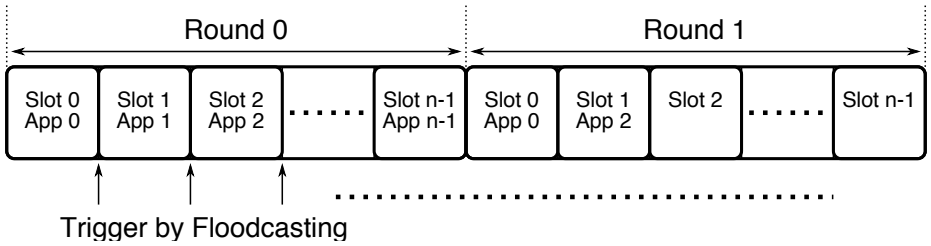


Figure 7.3: Super frame structure. Each round is divided into many slots, and each slot serves 1 application. All the slots are tightly synchronized across smart luminaires.

Concurrent applications are serviced using a super frame structure. A super frame is divided into many slots and each slot serves single application. Figure 7.3 shows the structure, a TDMA-based design. Applications can only request the lights at *round* level, which slot they are allocated within a round is determined by the scheduler on luminaire. An application will not be served twice within the same round. Since the applications have no knowledge of each other, the server APIs provide feedback to applications, namely whether it is successfully scheduled, scheduled with potential conflict, or failed to scheduled. The server calculates the schedulability based on the request round, number of slots in a round, slot length, and application length (number of consecutive slots required).

Once the synchronization packet arrives at a luminaire, the luminaire scheduler checks its internal data structure to see if any application should be run. The data is stored in a list sorted by transmit round and priority. Same-priority applications are scheduled using a *first come first serve* policy. Since applications may occupy more than one slot and we assume the application data is not dividable, the scheduler is designed to be a fixed priority non-preemptive scheduler [41]. Once the optical transmission starts, it guarantees the transmission cannot be interrupted by another application, even if the new application has higher priority, until the initial transmission is completed. If an application misses a round, it is not automatically rescheduled.

One simple way to resolve superframe capacity conflicts is to introduce dynamic slot length (time) and dynamic round capacity. The server could change the slot length to accommodate long application messages by modifying the synchronizing Floodcasting packets. This would prevent the scheduler from dropping applications resulting from conflicts. The length of the slot cannot be completely arbitrary, however. It must be greater than the length of application (T_{App}) plus the worst case latency of handling the interrupt (T_{Int}) and the time required to load the application into the optical TX fifo (T_{Load}) to guarantee the data is ready to transmit before next slot.

$$T_{slot} \geq T_{App} + T_{Int} + T_{Load}$$

On the other hand, a dynamic round capacity changes number of slots in a round. This approach trades reliable round latency for a greater number of concurrent applications.

7.3.2 Directions for Concurrency Support

In this section, we provide some guidance to VLC system designer for designing a communication protocol that is dividable, has enough redundancy, and agrees some global standard. A protocol with these properties supports better for concurrent applications.

Dividable Applications. One alternative to prevent dropping applications due to capacity conflicts is to divide long applications. If applications can be divided into non-adjacent slots, the single-round capacity conflict could be alleviated. A side effect of dividing applications is increasing the total overhead and complexity of the communication protocol, requiring state to be preserved across superframes. Additionally, $T_{Int} + T_{Load}$ would become non-negligible for extremely short slot lengths.

Error Detection/Correction. One problem in supporting frequency multiplexed applications is *aliasing* [82]. Lower sampling rate receivers may misinterpret data designated for a high rate receiver. Hence, applications must incorporate enough redundancy for error detection or even error correction. Checksums and cyclic redundancy checks (CRC) are commonly used techniques for detecting received data integrity. Moreover, erasure codes (ex. Fountain code [12]) can be used to reconstruct complete data from partially received chunks, further improving reliability.

Service Advertisement A systematic multiplexing solution could use designated slots for service advertisements. The advertisement packet could contain application IDs, their slot allocation within the round, and other round information. All receivers could listen to the advertisement packet and decide when to wakeup to receive the data. Service advertisements could provide a better energy saving scheme for the receiver but they requires all receivers to agree on a specific protocol for the advertisement slot and support a sufficient sense of local time to honor their slot offset.

CHAPTER 8

Conclusion and Future Work

LED lighting is becoming a popular source of indoor and outdoor illumination due in part to the long service life and high energy efficiency of LED light bulbs. LED light bulbs introduce new functionality to popular lighting, namely the ability to switch quickly and control the color of illumination. We explore how these new behaviors can be used to form a communication channel and present an full system architecture to enable new opportunities for post illumination applications.

8.1 Conclusion

In this dissertation, we present a new lighting architectural platform—*Software-Defined Lighting (SDL)*—that enables research into post-illumination applications of lighting. We explore the architectural trade-offs of SDL and present and implement a modular design of smart luminaire, which is composed of a LED frontend, control plane, and data plane. We demonstrate new applications enabled by our system including sub-meter accuracy indoor localization, optical wakeup, device configuration and programming, time synchronization of smart dust sensors, and camera communication to mobile devices.

We investigate how to use SDL to optically communicate with two types of receivers– diffusing photodiode and camera imager. Diffusing receivers absorb light from a wide field of view through the use of a photodiode. They are most commonly found on tiny smart dust sensors or receivers with a high data rate requirement. To avoid interference experienced by diffusing receivers from multiple smart luminaire transmitters, we present *Floodcasting*, a new wireless communication scheme based on concurrent transmissions. By synchronizing smart luminaires, sensor nodes with diffusing receivers can be woken up simultaneously, reducing the energy cost of neighbor discovery and idle listening for the potentially highly power constrained sensor nodes. Additionally, Floodcasting increases the coverage of each smart luminaire by creating constructive interference.

The second type of receiver we investigate is the camera imager. We leverage the CMOS camera available on nearly all smartphones, tablets and, smart glasses to receive data from smart luminaires by exploiting the *rolling shutter* effect. We explore visible light communication from a smart luminaire to a CMOS camera at a rate faster than is perceptible by the human eye using both frequency and Manchester encoded signals. Because a camera receiver is able to capture multiple smart luminaire transmitters in a single image, we are able to determine the angle-of-arrival (AoA). Using the AoA, we achieve a fine-grained positioning and orientation method for indoor localization. We present Luxapose, a AoA-based indoor localization that uses off-the-shelf mobile phones. Luxapose is the first VLCP system that is capable of decimeter-level accuracy with orientation information using unmodified smartphones and slightly modified commercial LED luminaires.

We believe that LED lighting will eventually dominate the indoor lighting market. This dissertation shows that the adoption of LED lighting also opens up many potential applications that might change the way we live. In this dissertation, we explore several key applications and propose one possible architecture for future lighting systems. This dissertation provides steps toward the realization of a future lighting infrastructure capable of supporting a myriad of new applications.

8.2 Future work

We envision several areas of future work including deploying software-defined lighting at a larger scale, refining the architecture, exploring alternate methods of synchronizing nearby luminaires, and optimizing the localization algorithms we employ.

8.2.1 Deploying in Large Scale

We present and implement the SDL architecture and demonstrate several new applications. Both the SDL and the applications we built have been tested only within a controlled, small scale environment. We plan to deploy the SDL into residential or commercial buildings. Such larger scale deployments would undoubtedly force a more complete and robust implementation of our architecture as well as expose any weaknesses we were not able to anticipate on the lab bench.

8.2.2 Refining the Architecture

We have demonstrated the modularized smart luminaire. Although we identify multiple data plane candidates, we primarily implement power line carrier and ethernet data plane.

At the time of this work, only one commercially available PLC development system was available.[1] The details of QCA7000 and its hardware design as well as the software libraries for the Stamp 1 are not open to the public, so we are unable to redesign a system to meet our target form factor. Furthermore, we plan on exploring other data plane options like WiFi.

8.2.3 Alternative Synchronization

In §4, we show a technique to achieve sub-µs time synchronization across a multi-hop wireless network using a flexible RF frontend. However, this technique requires hardware packet forwarding, a feature that does not exist in current commercial radios. Meanwhile, purely software implementations cannot easily provide the necessary level of synchronization due to processing time jitter in the software. Therefore, it is challenging to tightly synchronize nodes using commercial radios. Possible alternative approaches use network connectivity provided by the data plane or other side channels to provide tight synchronization. Also, the multi-hop constraint no longer holds for PLC or Ethernet data planes in the same way that they hold for low-power wireless protocols like 802.15.4. Since PLC is a shared medium, in principle, data could be received by all luminaires on the PLC network simultaneously. Hence, it might be possible to time synchronize nodes using a broadcast message, but it would require low-level access to the PLC transceivers. On the other hand, if Ethernet is used for the data plane, IEEE 1588 precision time protocol (PTP) may be a promising candidate for time synchronization. PTP achieves clock synchronization accuracy in sub-µs range, and many commercial products are available today [3].

8.2.4 Optimizing Localization Algorithms

Indoor localization is a promising application of SDL. Luxapose provides decimeter level accuracy and orientation information without require custom receivers. Our implementation heavily relies on the cloud/cloudlet server to process images and estimate locations. The latency of the entire process is mainly dominated by capturing a 33 MP image and uploading the high resolution image to the server. The latency may be critical in any real environment with poor wireless connection.

Reducing the image processing latency is another obvious direction for improvement. The simplest way to reduce the overall latency is to reduce the image resolution. But reducing the image resolution degrades the location accuracy, and increases the rolling shutter

[1] The PLC Stamp 1 from I2SE, which features a Qualcomm QCA7000 HomePlug Green PHY and a Freescale MK20D microcontroller [39]

scan rate of the camera. This is the trade-off between location accuracy, total latency, and location update rates. These parameters need to be characterized to layout the design spaces for different application requirements. For instance, indoor navigation in supermarkets require lower accuracy but faster response, whereas the precision robot navigation in a warehouse requires higher accuracy but longer latency tolerant.

Another possibility for reducing latency is to perform some local computation. This reduces the amount of data transferred to the cloud/cloudlet server, and potentially speeds up the process of determining a location. Since the computational resources on the mobile devices are limited, careful partitioning of the computation between mobile devices and cloud/cloudlet server is necessary to ensure overall gain. Compressed sensing might be an interesting area to explore. Since the majority of an image contains no information, the size of image could be significantly reduced if it were possible to intelligently sub-sample the image. However, this approach requires camera support which is unavailable today.

In addition, many research questions are enabled by Luxapose. For example, what is the best-suited image processing techniques for these extremely-low exposure images? What is the local image processing boundary to reduce the data and further reduce the latency? What type of computation can be quickly processed by mobile devices? What is the energy/latency trade-off between local and cloud/cloudlet processing? Furthermore, other architecture/deployment questions include how to automatically allocate the locations of lighting fixtures given a blueprint so that any location can be covered by at least 3 transmitters? How to allocate the frequency of each transmitter to maximize the channel reuse? What is the best side channel to encode coordinates of transmitters? How to integrate other localization techniques (ex. RSSI, inertial navigation) to provide better coverage? These are interesting questions/areas that are worth exploring to provide a better localization system.

APPENDIX A

Design Files

Most of the design files (Software, Schematic, and PCBs) can be downloaded from our lab website, except the PLC data plane from I2SE (I2SE Stamp 1 PLC modem), which is a closed source hardware.

Smart Luminaire http://lab11.eecs.umich.edu/projects/vlc_localization/

Dev. Platform http://lab11.eecs.umich.edu/projects/usdr/

VLCP http://lab11.eecs.umich.edu/projects/vlc_localization/

I2SE PLC http://www.i2se.com/powerline.html

BIBLIOGRAPHY

[1] Arborlight. http://www.arborlight.com/.

[2] Bytelight. http://www.bytelight.com/.

[3] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pages c1–269, July 2008.

[4] IEEE Standard for Information technology– Local and metropolitan area networks–Specific requirements–Part 3: CSMA/CD Access Method and Physical Layer Specifications Amendment 3: Data Terminal Equipment (DTE) Power via the Media Dependent Interface (MDI) Enhancements. *IEEE Std 802.3at-2009 (Amendment to IEEE Std 802.3-2008)*, pages 1–137, Oct 2009.

[5] IEEE Standard for Local and Metropolitan Area Networks–Part 15.7: Short-Range Wireless Optical Communication Using Visible Light. *IEEE Std 802.15.7-2011*, pages 1–309, Sept 2011.

[6] M. Z. Afgani, H. Haas, H. Elgala, and D. Knipp. Visible light communication using ofdm. In *Proceedings of the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom '06)*, pages 6–11. IEEE, 2006.

[7] J. Armstrong, Y. A. Sekercioglu, and A. Neild. Visible light positioning: A roadmap for international standardization. *IEEE Communications Magazine*, 51(12):2–7, 2013.

[8] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings 19th Annual Joint Conference of the IEEE Computer and Communications Societies. (INFOCOM '00)*, volume 2, pages 775–784, 2000.

[9] S. M. Berman, D. S. Greenhouse, I. L. Bailey, R. D. Clear, and T. W. Raasch. Human electroretinogram responses to video displays, fluorescent lighting, and other high frequency sources. *Optometry & Vision Science*, 68(8):645–662, 1991.

[10] D. B. Boivin, J. F. Duffy, R. E. Kronauer, and C. A. Czeisler. Dose-response relationships for resetting of human circadian clock by light. 1996.

[11] C. Branas, F. Azcondo, and J. Alonso. Solid-State Lighting: A System Review. *IEEE Industrial Electronics Magazine*, 7(4):6–14, 2013.

[12] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. *ACM SIGCOMM Computer Communication Review*, 28(4):56–67, 1998.

[13] A. Cailean, B. Cagneau, L. Chassagne, S. Topsu, Y. Alayli, and J.-M. Blosseville. Visible light communications: Application to cooperation between vehicles and road infrastructures. In *IEEE Intelligent Vehicles Symposium (IV '12)*, pages 1055–1059. IEEE, 2012.

[14] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 679–698, Nov 1986.

[15] CASIO. Picapicamera. http://www.casio-isc.com/en/.

[16] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proceedings of the 16th ACM Annual International Conference on Mobile Computing and Networking (MobiCom '10)*, pages 173–184, 2010.

[17] T. E. Cohn. Integration by the human eye: Implications for warning signal design. *University of California Transportation Center*, 1993.

[18] CommScope. Five case studies reveal the potential of saving energy with intelligent lighting network solutions.

[19] P. Connolly and D. Bonte. Indoor location in retail: Where is the money?, 2013.

[20] G. Cossu, A. Khalid, P. Choudhury, R. Corsini, E. Ciaramella, et al. 3.4 Gbit/s visible optical wireless transmission based on RGB LED. *Optics express*, 20(26):B501–B506, 2012.

[21] CREE. Cree® XLamp® XM-L Color LEDs datasheet. http://www.cree.com/~/media/Files/Cree/LED20Components20and20Modules/XLamp/Data20and20Binning/XLampXML_Color.pdf.

[22] K. Cui, G. Chen, Z. Xu, and R. D. Roberts. Line-of-sight visible light communication system design and demonstration. *7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP '10)*, pages 621–625, 2010.

[23] C. A. Czeisler. The effect of light on the human circadian pacemaker. *Circadian clocks and their adjustment*, 254:302, 1995.

[24] C. Danakis, M. Afgani, G. Povey, I. Underwood, and H. Haas. Using a CMOS camera sensor for visible light communication. In *IEEE Globecom Workshops*, pages 1244–1248, 2012.

[25] M. C. Dean. *Bearings-Only Localization and Mapping*. PhD thesis, Carnegie Mellon University, 2005.

[26] P. Dietz, W. Yerazunis, and D. Leigh. Very low-cost sensing and communication using bidirectional LEDs. In *UbiComp 2003: Ubiquitous Computing*, pages 175–191. Springer, 2003.

[27] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis. Design and evaluation of a versatile and efficient receiver-initiated link layer for low power wireless. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*, pages 1–14. ACM, 2010.

[28] P. Dutta, Y.-S. Kuo, A. Ledeczi, T. Schmid, and P. Volgyesi. Putting the software radio on a low-calorie diet. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets '10)*, pages 20:1–20:6. ACM, 2010.

[29] P. Dutta, R. Musăloiu-E, I. Stoica, and A. Terzis. Wireless ack collisions not considered harmful. In *Proceedings of the 7th ACM SIGCOMM Workshop on Hot Topics in Networks 2008*, (Hotnets '08).

[30] A. El-Osery and C. Abdallah. Power control in cdma cellular systems. http://www.eetimes.com/document.asp?doc_id=1275804.

[31] Electric Imp. Electric Imp datasheet. https://electricimp.com/docs/attachments/hardware/datasheets/imp001_specification.pdf.

[32] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele. Low-power wireless bus. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys '12)*, pages 1–14. ACM, 2012.

[33] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with glossy. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN '11)*, pages 73–84. IEEE, 2011.

[34] N. C. Greenham, R. H. Friend, and D. D. Bradley. Angular dependence of the emission from a conjugated polymer light-emitting diode: Implications for efficiency calculations. *Advanced Materials*, 6(6):491–494, 1994.

[35] L. Grobe, A. Paraskevopoulos, J. Hilt, D. Schulz, F. Lassak, F. Hartlieb, C. Kottke, V. Jungnickel, and K.-D. Langer. High-speed visible light communication systems. *IEEE Communications Magazine*, 51(12):60–66, 2013.

[36] J. Grubor, S. Randel, K.-D. Langer, and J. W. Walewski. Broadband information broadcasting using LED-based interior lighting. *Journal of Lightwave technology*, 26(24):3883–3892, 2008.

[37] R. Haitz and J. Y. Tsao. Solid-state lighting:'the case'10 years after and future prospects. *physica status solidi (a)*, 208(1):17–29, 2011.

[38] T. Hao, R. Zhou, and G. Xing. Cobra: color barcode streaming for smartphone systems. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*, pages 85–98. ACM, 2012.

[39] I2SE. I2SE PLC-Stamp 1 datasheet. http://www.cree.com/~/media/Files/Cree/LED20Components20and20Modules/XLamp/Data20and20Binning/XLampXML_Color.pdf.

[40] International Commission on Illumination. Commission internationale de l'eclairage proceedings. 1931.

[41] K. Jeffay, D. F. Stanat, and C. U. Martel. On non-preemptive scheduling of period and sporadic tasks. In *Proceedings of the 12th Real-Time Systems Symposium*, pages 129–139. IEEE, 1991.

[42] A. Jovicic, J. Li, and T. Richardson. Visible light communication: Opportunities, challenges and the path to market. *IEEE Communications Magazine*, 51(12):26–32, 2013.

[43] G. Kim, Y. Lee, S. Bang, I. Lee, Y. Kim, D. Sylvester, and D. Blaauw. A 695 pw standby power optical wake-up receiver for wireless sensor nodes. In *Custom Integrated Circuits Conference (CICC '12)*, pages 1–4. IEEE, 2012.

[44] D. F. Kirill Serkh, Olivia Walch. Entrain. http://entrain.math.lsa.umich.edu/.

[45] T. Komine and M. Nakagawa. Integrated system of white LED visible-light communication and power-line communication. *IEEE Transactions on Consumer Electronics*, 49(1):71–79, Feb 2003.

[46] T. Komine and M. Nakagawa. Fundamental analysis for visible-light communication system using LED lights. *IEEE Transactions on Consumer Electronics*, 50(1):100–107, 2004.

[47] R. Kraemer and M. Katz. *Short-range wireless communications: emerging technologies and applications*. Wiley, 2009.

[48] A. Krioukov, S. Dawson-Haggerty, L. Lee, O. Rehmane, and D. Culler. A living laboratory study in personalized automated lighting controls. In *Proceedings of the 3rd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys '11)*, pages 1–6. ACM, 2011.

[49] Y.-S. Kuo, P. Pannuto, and P. Dutta. System architecture directions for a software-defined lighting infrastructure. In *Proceedings of the 1st ACM MobiCom Workshop on Visible Light Communication Systems*, VLCS '14, pages 3–8. ACM, 2014.

[50] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking (MobiCom '14)*. ACM, 2014.

[51] Y.-S. Kuo, P. Pannuto, T. Schmid, and P. Dutta. Reconfiguring the software radio to improve power, price, and portability. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys '12)*. ACM, 2012.

[52] K.-D. Langer, J. Vucic, C. Kottke, L. Fernández, K. Habe, A. Paraskevopoulos, M. Wendl, and V. Markov. Exploring the potentials of optical-wireless communication using white LEDs. In *Proceedings of the 13th International Conference on Transparent Optical Networks (ICTON '11)*, pages 1–5. IEEE, 2011.

[53] H. Le-Minh, D. O'brien, G. Faulkner, L. Zeng, K. Lee, D. Jung, Y. Oh, and E. T. Won. 100-Mb/s NRZ visible light communications using a postequalized white LED. *Photonics Technology Letters, IEEE*, 21(15):1063–1065, Aug 2009.

[54] Y. Lee, S. Bang, I. Lee, Y. Kim, G. Kim, M. H. Ghaed, P. Pannuto, P. Dutta, D. Sylvester, and D. Blaauw. A modular 1 mm^3 die-stacked sensing platform with low power I^2C inter-die communication and multi-modal energy harvesting. *IEEE Journal of Solid-State Circuits*, 48(1):229–243, 2013.

[55] Y. Lee, B. Giridhar, Z. Foo, D. Sylvester, and D. Blaauw. A Sub-nW Multi-stage Temperature Compensated Timer for Ultra-Low-Power Sensor Nodes. *IEEE Journal of Solid-State Circuits (JSSC '13)*, 2013.

[56] Y. Lee, G. Kim, S. Bang, Y. Kim, I. Lee, P. Dutta, D. Sylvester, and D. Blaauw. A modular 1mm^3 die-stacked sensing platform with optical communication and multi-modal energy harvesting. In *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC '12)*, pages 402–404. IEEE, 2012.

[57] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao. Epsilon: A visible light based positioning system. In *Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI '14)*, pages 331–344, 2014.

[58] T. Li, C. An, A. Campbell, and X. Zhou. Hilight: hiding bits in pixel translucency changes. In *Proceedings of the 1st ACM MobiCom workshop on Visible light communication systems*, VLCS '14, pages 45–50. ACM, 2014.

[59] R. LiKamWa, D. Ramirez, and J. Holloway. Styrofoam: A tightly packed coding scheme for camera-based visible light communication. In *Proceedings of the 1st ACM MobiCom Workshop on Visible Light Communication Systems*, VLCS '14, pages 27–32. ACM, 2014.

[60] S.-K. Lim, K. Ruling, I. Kim, and I. Jang. Entertainment lighting control network standardization to support VLC services. *IEEE Communications Magazine*, 51(12):42–48, 2013.

[61] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297. University of California Press, 1967.

[62] P. R. Mills, S. C. Tomkins, and L. J. Schlangen. The effect of high correlated colour temperature office lighting on employee wellbeing and work performance. *Journal of Circadian Rhythms*, 5(1):2, 2007.

[63] Navigant Consulting. 2010 U.S. Lighting Market Characterization. *U.S. Department of Energy*, 2012.

[64] Navigant Consulting. Energy savings potential of solid-state lighting in general illumination applications. *U.S. Department of Energy*, 2012.

[65] netatmo. JUNE. https://www.netatmo.com/en-US/product/specifications/june.

[66] H. Noguchi and T. Sakaguchi. Effect of illuminance and color temperature on lowering of physiological activity. *Applied human science*, 18(4):117–123, 1999.

[67] Nokia. White paper Nokia Lumia 1020. http://http://i.nokia.com/blob/view/-/2723846/data/.../-/Lumia1020-whitepaper.pdf.

[68] M. Noshad and M. Brandt-Pearce. Can visible light communications provide Gb/s service? *arXiv preprint arXiv:1308.3217*, 2013.

[69] D. O'brien, L. Zeng, H. Le-Minh, G. Faulkner, J. Walewski, and S. Randel. Visible light communications: Challenges and possibilities. In *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '08)*, pages 1–5, Sep 2008.

[70] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[71] R. Pickholtz, D. Schilling, and L. Milstein. Theory of spread-spectrum communications–a tutorial. *IEEE Transactions on Communications*, 30(5):855–884, May 1982.

[72] N. Rajagopal, P. Lazik, and A. Rowe. Hybrid visible light communication for cameras and low-power embedded devices. In *Proceedings of the 1st ACM MobiCom workshop on Visible light communication systems*, VLCS '14, pages 33–38. ACM, 2014.

[73] N. Rajagopal, P. Lazik, and A. Rowe. Visual light landmarks for mobile devices. In *Proceedings of the 13th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '14)*, pages 249–260, 2014.

[74] S. Rajagopal, R. D. Roberts, and S.-K. Lim. IEEE 802.15. 7 visible light communication: Modulation schemes and dimming support. *IEEE Communications Magazine*, 50(3):72–82, March 2012.

[75] M. S. Rea. The IESNA lighting handbook: reference & application. 2000.

[76] A. Richardson, J. Strom, and E. Olson. AprilCal: Assisted and repeatable camera calibration. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS '13)*, pages 1814–1821, 2013.

[77] M. Sakata, Y. Yasumuro, M. Imura, Y. Manabe, and K. Chihara. Location system for indoor wearable PC users. In *Workshop on Advanced Computing and Communicating Techniques for Wearable Information Playing*, pages 57–61, 2003.

[78] X. san Yin and C.-N. Shen. The study of aging frequency drift mechanism for quartz crystal resonators. In *Symposium on Piezoelectricity, Acoustic Waves and Device Applications (SPAWDA '13)*, pages 1–4, Oct 2013.

[79] M. Satyanarayanan, G. Lewis, E. Morris, S. Simanta, J. Boleng, and K. Ha. The role of cloudlets in hostile environments. *IEEE Pervasive Computing*, 12(4):40–49, 2013.

[80] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross. LED-to-LED visible light communication networks. In *Proceedings of the 14th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '13)*, pages 1–10. ACM, 2013.

[81] T. Schmid. GNU radio 802.15. 4 en-and decoding. *UCLA NESL, Los Angeles, CA*, 2005.

[82] C. E. Shannon. Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers (IRE)*, 37(1):10–21, 1949.

[83] J. Tan and N. Narendran. A driving scheme to reduce AC LED flicker. *Optical Engineering*, 2013.

[84] U.S. Congress. Energy independence and security act of 2007. *Public Law*, 2, 2007.

[85] U.S. Department of Energy. 2010 Commercial energy end-use splits, by fuel type. *Building Energy Data Book*, 2012.

[86] A. J. Viterbi. *CDMA: Principles of Spread Spectrum Communication*. Addison Wesley Longman Publishing Co., Inc., 1995.

[87] A. Voglgsang, T. Langguth, G. Korner, H. Steckenbiller, and R. Knorr. Measurement, characterization and simulation of noise on powerline channels. In *Proceedings of the 4th International Symposium on Power Line Communications and Its Applications (ISPLC '00)*, pages 139–146, 2000.

[88] J. Vučić, C. Kottke, S. Nerreter, K.-D. Langer, and J. W. Walewski. 513 Mbit/s visible light communications link based on dmt-modulation of a white LED. *Journal of Lightwave Technology*, 28(24):3512–3518, 2010.

[89] P. Waide and S. Tanishima. *Light's Labour's Lost: Policies for Energy-efficient Lighting*. OECD Publishing, 2006.

[90] Q. Wang, D. Giustiniano, and D. Puccinelli. Openvlc: Software-defined visible light embedded networks. In *Proceedings of the 1st ACM MobiCom Workshop on Visible Light Communication Systems*, VLCS '14. ACM, 2014.

[91] A. Wilkins, J. Veitch, and B. Lehman. Led lighting flicker and potential health concerns: Ieee standard par1789 update. In *IEEE Energy Conversion Congress and Exposition (ECCE '10)*, pages 171–178, Sep. 2010.

[92] M. A. Youssef and A. Agrawala. The Horus WLAN location determination system. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, pages 205–218, 2005.

[93] S.-H. Yu, O. Shih, H.-M. Tsai, and R. Roberts. Smart automotive lighting for vehicle safety. *IEEE Communications Magazine*, 51(12):50–59, 2013.